

Adaptive Credit Scoring using Local Classification Methods

Dimitris Nikolaidis

School of Production Engineering and Management, Technical University of Crete

2022

A thesis submitted in fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Τίτλος (ελληνικά/αγγλικά):

Προσαρμοστικά Μοντέλα Πιστοληπτικής Αξιολόγησης Μέσω Τοπικών Μεθόδων Ταξινόμησης

Adaptive Credit Scoring Using Local Classification Methods

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτρης Νικολαΐδης

ΤΡΙΜΕΛΗΣ ΣΥΜΒΟΥΛΕΥΤΙΚΗ ΕΠΙΤΡΟΠΗ:

1. Καθηγητής Μιχάλης Δούμπος (Επιβλέπων)
2. Καθηγητής Κωνσταντίνος Ζοπουνίδης
3. Καθηγητής Φώτιος Πασιούρας

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την: 11 / 1 / 2023

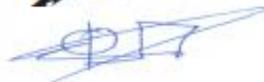
1. Καθηγητής Μιχάλης Δούμπος (Πολυτεχνείο Κρήτης)



2. Καθηγητής Κωνσταντίνος Ζοπουνίδης (Πολυτεχνείο Κρήτης)



3. Καθηγητής Φώτιος Πασιούρας (Montpellier Business School)



4. Καθηγητής Ευάγγελος Γρηγορούδης (Πολυτεχνείο Κρήτης)



5. Καθηγητής Χρυσοβαλάντης Γαγάνης (Πανεπιστήμιο Κρήτης)



6. Αναπληρωτής Καθηγητής Γεώργιος Ατσαλάκης (Πολυτεχνείο Κρήτης)



7. Αναπληρωτής Καθηγητής Χρήστος Λεμονάκης (Ελληνικό Μεσογειακό Πανεπιστήμιο)



E

Το πρόβλημα της ασυμμετρίας της πληροφορίας (information asymmetry) έχει μελετηθεί εκτενώς όπως και οι συνέπειές του στο χρηματοπιστωτικό χώρο. Έτσι η ανταλλαγή πληροφοριών και δεδομένων οικονομικής συμπεριφοράς, μέσω μηχανισμών όπως τα γραφεία πίστης (Credit bureaus) λειτουργούν ως αντίβαρο στην ασυμμετρία αυτή και ως υποστηρικτικό εργαλείο στις πιστοδοτικές αποφάσεις. Από το τα τέλη του 19ου αιώνα που λειτουργήσε το πρώτο γραφείο πίστης Dun & Bradstreet, αναπτύχθηκαν μεθοδολογίες για την υποστήριξη της πιστοληπτικής αξιολόγησης υποψηφίων δανειοληπτών. Η βασικότερη ίσως μεθοδολογία των γραφείων πίστης ιστορικά είναι η πιστοληπτική βαθμολόγηση και συνίσταται στη χρήση στατιστικών και αλγοριθμικών μεθόδων που αποσκοπούν στο μετασχηματισμό των δεδομένων σε αριθμητικές μετρήσεις οι οποίες μπορούν να χρησιμοποιηθούν για την αυτοματοποιημένη "κατάρτιση προφίλ" υποψηφίων δανειοληπτών. Μεθοδολογικά η πιστοληπτική βαθμολόγηση αρχικά στηρίζονταν σε αμιγώς στατιστικές προσεγγίσεις (π.χ. λογιστική παλινδρόμηση, δέντρα αποφάσεων κλπ), ωστόσο η σχετικά πρόσφατη "έκρηξη" των μεθόδων μηχανικής μάθησης (machine) οδήγησε σε αντίστοιχη ανάπτυξη των σχετικών μεθόδων και υποδειγμάτων που χρησιμοποιούνται στην πιστωτική βαθμολόγηση.

Παρόλα αυτά η εφαρμογή αυτών των μεθόδων συναντά θεωρητικά αλλά και πρακτικά προβλήματα, το βασικότερο των οποίων είναι η πληθυσμιακή μετατόπιση (population πιστοληπτικής βαθμολόγησης αντιμετωπίζουν το πρόβλημα της πληθυσμιακής μετατόπισης (population drift), όταν οι στατιστικές κατανομές του υπό μοντελοποίηση πληθυσμού, αναπόφευκτα, μεταβάλλονται στο χρόνο. Αυτό το πρόβλημα αντιμετωπίζεται

με τη διαρκή παρακολούθηση (Monitoring) των επιδόσεων των υποδειγμάτων λαμβάνοντας υπόψη το γεγονός ότι για την ανάπτυξη τέτοιων μοντέλων χρειάζονται δεδομένα κατ' ελάχιστο 2 ετών και προθέτοντας και τον απαιτούμενο χρόνο υλοποίησης και θέσης σε παραγωγική λειτουργία, σε πρακτικό επίπεδο εντείνεται ακόμα περισσότερο το πρόβλημα της πληθυσμιακής μετατόπισης.

Στην παρούσα διατριβή προτείνεται η αντιμετώπιση του προβλήματος της πληθυσμιακής μετατόπισης με αυτόματη και δυναμική προσαρμογή των υποδειγμάτων βαθμολόγησης με χρήση τοπικών μεθόδων ταξινόμησης (local classification). Συγκεκριμένα το προτεινόμενο σχήμα συνίσταται στον υπολογισμό της πιστοληπτικής βαθμολόγησης χρησιμοποιώντας μεθόδους Lazy learning για κάθε ένα εισερχόμενο αίτημα score (σημείο εισόδου ή query instance), χρησιμοποιώντας μόνο εκείνο το υποσύνολο των ομοειδών εγγραφών προς το εισερχόμενο σημείο (Instance selection, local region of competence). Η έννοια της ομοιότητας (similarity) καθορίζεται από την απόσταση (distance) με συγκεκριμένη μετρική (π.χ. ευκλείδια απόσταση) μεταξύ της εισερχόμενης εγγραφής και του n -διάστατου χώρου του συνόλου των εγγραφών (feature space), όπου είναι το πλήθος των διαφορετικών μεταβλητών (attributes ή characteristics), όπου n είναι το πλήθος των πεδίων κάθε εγγραφής. Το υποσύνολο των ομοειδών εγγραφών κάθε εισερχόμενου σημείου προσδιορίζεται με τη μέθοδο των πλησιέστερων γειτόνων (kNN). Έτσι κάθε γειτονιά χρησιμοποιείται ως σύνολο εκπαίδευσης (training set) ενός υποδείγματος πιστωτικής βαθμολόγησης αποκλειστικά για το συγκεκριμένο σημείο εισόδου.

Συγκρίνονται μεθοδολογίες στατιστικές και μηχανικής μάθησης (λογιστική παλινδρόμηση που λαμβάνεται και ως σημείο αναφοράς, Random Forests και Gradient Boosting Trees), χρησιμοποιώντας πραγματικά δεδομένα γραφείου πίστης για ένα βάθος 11 ετών (2009-2019) ανά τρίμηνο με συνολικά 3,520,000 εγγραφές και 125 διαφορετικές μεταβλητές. Για τον υπολογισμό των μέτρων επίδοσης (performance measures) χρησιμοποιήθηκαν τα AUC Measure με κατάλληλες στατιστικές μεθοδολογίες σύγκρισης διαφορετικών ταξινομητών Friedman's aligned ranks σε συνδυασμό με το post-hoc Nemenyi test

Ειδικότερα διερευνήθηκαν οι εξής στατιστικές υποθέσεις:

H

Έχουν καλύτερες επιδόσεις οι τοπικές μέθοδοι (local classification methods) σε σχέση με Ης καθολικές (global);

Υπάρχει σημαντικά στατιστική διαφοροποίηση μεταξύ των μεθόδων μάθησης και της λογιστικής παλινδρόμησης?

Επιπλέον, επιλέχθηκε η καλύτερη τοπική μέθοδος με βάση την ομοιότητα (KNN) και οι μέθοδοι μαρτυρούν κατά περίπτωση καλύτερα αποτελέσματα σε σχέση με τις καθολικές, ωστόσο η διαφορά είναι στατιστικά σημαντικές μόνο στην περίπτωση της λογιστικής παλινδρόμησης. Ιδιαίτερα ενδιαφέρον παρουσιάζει το γεγονός ότι, σε συμφωνία με τα ευρήματα της βιβλιογραφίας, οι μέθοδοι μηχανικής μάθησης που εφαρμόστηκαν είναι περίπου 6%-7% καλύτερες (με μετρική AUC) σε σχέση με την καθολική λογιστική παλινδρόμηση, ωστόσο η τοπική λογιστική παλινδρόμηση βρίσκεται περίπου στο ίδιο επίπεδο επιδόσεων με τις μεθόδους μηχανικής μάθησης. Τέλος η επιλογή γειτόνων με βάση την ομοιότητα ως προς

το σημείο εισόδου αποδεικνύεται ότι φέρει σημαντική βελτίωση στην επίδοση, σε σχέση με την επιλογή τυχαίων σημείων χωρίς να λαμβάνεται υπόψη η γειτνίαση.

Abstract

Despite the advances in machine learning methods which are also applied in credit scoring with overall 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 machine learning methods (random forests and gradient boosting trees) that have shown promising results in the literature machine learning). Our data sample has been derived from a proprietary credit bureau database and spans a period of 11 consequent years with quarterly sampling frequency consisting of more than 3,520,000 record-month observations. Rigorous performance measures used in credit scoring literature and practice (such as AUROC and 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 machine learning ones which are considered “black box” methods.

Keywords: concept/population drift; adaptive learning; local classification; behavioral credit scoring; lazy learning; region of competence

Table of Contents

| | | |
|---|---|----|
| T | Abstract..... | 7 |
| O | 1 Information Asymmetry, Credit Bureaus and Credit Scoring..... | 15 |
| C | 1.1 Significance of the study..... | 22 |
| | 2 Background and Related Theoretical Framework | 25 |
| \ | 2.1 Credit Scoring..... | 25 |
| o | 2.1.1 Credit Scoring Formalization..... | 29 |
| | 2.1.2 Recent Advances..... | 29 |
| " | 2.1.3 Challenges and Issues | 35 |
| 1 | 2.2 Concept Drift and Adaptive Learning..... | 41 |
| - | 2.3 Adaptive credit scoring..... | 44 |
| 3 | 2.4 Local Classification | 47 |
| " | 2.5 Local Regions of Competence | 50 |
| | 2.6 Imbalanced Classification..... | 52 |
| \ | 2.7 Methodological issues in classifiers' performance measures and h comparisons | 54 |
| | 2.7.1 Performance measures | 54 |
| \ | 2.7.2 Comparison of Classifiers..... | 56 |
| z | 3 Experimental Setup and Methodology..... | 59 |
| | 3.1 Problem Formulation | 59 |
| \ | 3.2 Data and Variables | 61 |
| u | 3.3 Scoring Parameters | 62 |

| | | |
|-------|--|-----|
| 3.4 | Methodology | 63 |
| 3.4.1 | Local Classification | 65 |
| 3.4.2 | Global Classification..... | 69 |
| 4 | Empirical Results | 71 |
| 4.1 | Global classifiers and Population drift..... | 71 |
| 4.2 | Local classification | 78 |
| 4.2.1 | Euclidean vs Mahalanobis Distance and size of k (kNN)..... | 78 |
| 4.2.2 | Local vs Global Classifiers | 81 |
| 4.3 | Random regions of competence vs kNNs..... | 87 |
| 5 | Conclusions and Future Work..... | 89 |
| | Appendix A: Alternative Data..... | 93 |
| | Appendix B: List of Variables..... | 94 |
| | Appendix C: Execution Environment..... | 102 |
| | Appendix D: Detailed Results | 106 |
| | References..... | 113 |

Table of Tables

| | |
|--|-----------|
| Table 1: Data coverage width of European Credit Registries (CRAs) (Source: ACCIS (2020), ACCIS Membership Survey 2020”, numbers above countries indicate numbers of bureaus per country) | 31 |
| Table 2: Alternative credit scoring products (Source: (Hurley & Adebayo, 2016) and author’s analysis) | 32 |
| Table 3: Performance measures of global classifiers (no retrain=shaded rows) over all snapshots with different feature selection mechanisms (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain) | 71 |
| Table 4: Performance measures of LR_L using different distance metrics and local region sizes (LR=Logistic Regression, 2k, 4k, 6k=k in kNN, euc=Euclidion dist. mah=Mahalanobis) | 78 |
| Table 5: Performance of Local vs Global Classifiers (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, , bold indicate the best classifier for the specific snapshot))..... | 83 |
| Table 6: Performance of Local vs Global Classifiers (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN) | 90 |
| <i>Table A-7: Alternative data in Credit Scoring Source: (ICCR, 2019)</i> | <i>93</i> |
| <i>Table A-8: Types of data used in Credit Scoring Source: (ICCR, 2019)</i> | <i>93</i> |
| Table A-9: Performance (AUC) for Global Classifiers LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, SD= Standard | |

deviation, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain)
 106

Table A-10: Performance (**H-Measure**) for Global Classifiers LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, SD= Standard deviation, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain)
 107

Table A-11: Comparison of different local region sizes (*kNNs*) using **Euclidean distance** (LR=Logistic Regression, L=Local classifier, 2k=2000, 4k=4000, 6k=6000 for kNN) 108

Table A-12: Comparison of different local region sizes (*kNNs*) using **Mahalanobis distance** (LR=Logistic Regression, L=Local classifier, 2k=2000, 4k=4000, 6k=6000 for kNN) 109

Table A-13: Local vs Global Classifiers (AUC) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, *= training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot).....110

Table A-14: Local vs Global Classifiers (H-Measure) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, *= training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot).....111

Table A-15: *kNNs* vs Random sub-sampling (LR=Logistic Regression, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, rnd=random, *= training snapshot for global classifiers)112

Table of Figures

| | |
|--|-----------|
| Figure 1: Performance gain of ML methods versus Logit (Source: (Alonso & Carbó, 2020) | 35 |
| Figure 2: An ontology of performance metrics (Source: (Japkowicz & Shah, 2011))..... | 55 |
| Figure 3: Statistical tests for comparing multiple classifiers (Source: (Japkowicz & Shah, 2011)) | 57 |
| Figure 4: Observation and Outcome windows | 59 |
| Figure 5: High-level flow for the proposed local classification scheme ($ S $ denotes the cardinality of a set S)..... | 66 |
| Figure 6: Training phase for the proposed local classification scheme ($ S $ denotes the cardinality of a set S) | 67 |
| Figure 7: Global classification scheme ($ S $ denotes the cardinality of a set S) | 69 |
| <i>Figure 8: AUC of global classifiers (y-axis not starting from zero) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain,).....</i> | <i>72</i> |
| <i>Figure 9: H-Measure of global classifiers (y-axis not starting from zero) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain,).....</i> | <i>73</i> |
| <i>Figure 10: AUC based statistical differences of global classifiers (p-value matrix) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)</i> | <i>74</i> |
| <i>Figure 11: H-measure statistical differences of global classifiers (p-value matrix) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)</i> | <i>75</i> |

| | |
|---|----|
| <i>Figure 12: Graph of rankings for global models (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)</i> | 76 |
| Figure 13: AUC degradation of global classifiers with and without retraining (y-axis not starting from 0) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, dashed lines=no retrain) | 77 |
| Figure 14: : H-Measure degradation of global classifiers with and without retraining (y-axis not starting from 0) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, dashed lines=no retrain) | 77 |
| <i>Figure 15: AUC of LR_L using different distance metrics and local region sizes (y-axis not starting from zero) (LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidion dist. maha=Mahalanobis)</i> | 79 |
| <i>Figure 16: H-Measure of LR_L using different distance metrics and local region sizes (y-axis not starting from zero) (LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidion dist. maha=Mahalanobis)</i> | 79 |
| <i>Figure 17: AUC based statistical differences of LR-L using different distance metrics and local region sizes (p-value matrix) (LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidion dist. maha=Mahalanobis)</i> | 80 |
| <i>Figure 18: H-Measure based statistical differences of LR-L using different distance metrics and local region sizes (p-value matrix) (LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidion dist. maha=Mahalanobis)</i> | 81 |
| Figure 19: Pairwise timeline comparison between local/global classifiers sizes (y-axis not starting from zero) (different y-axis scales, LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, solid blue line denotes local classifier, red line with markers global classifier) | 82 |

| | |
|--|----|
| <i>Figure 20: AUC of Local vs Global Classifiers (y-axis not starting from zero) LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN).....</i> | 83 |
| <i>Figure 21: AUC of Local vs Global Classifiers (y-axis not starting from zero) LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN).....</i> | 84 |
| <i>Figure 22: Graph of rankings for global models (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.).....</i> | 84 |
| <i>Figure 23: AUC based statistical differences of Local vs Global Classifiers (p-value matrix)...</i> | 85 |
| <i>Figure 24: H-Measure based statistical differences of Local vs Global Classifiers (p-value matrix)</i> | 86 |
| <i>Figure 25: Critical Distances between local and global classifiers (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN)</i> | 87 |
| <i>Figure 26: kNNs vs random regions (different y-axis scales) (LR=Logistic Regression, G=Global Classifier, 2k=2000 for kNN, *= training snapshot for global LR)</i> | 88 |
| <i>Figure 27: Statistical differences of kNNs vs random regions</i> | 88 |

1 Information Asymmetry, Credit Bureaus and Credit Scoring

In economic theory information asymmetry has far reaching and well-studied consequences in the operation of financial markets. According to (Akerlof, 1978), whose work on information asymmetry is among the oldest and best known, when only the average quality of the good can be assumed in markets with a good of indeterminate quality, over time goods of above-average quality will be driven out and will threaten the viability of the market for the good. In lending, the problem of asymmetric information stems from the fact that a lender's knowledge of a borrower's likelihood to repay (their "risk profile") is imprecise and must be inferred based upon available information. Thus in the case of consumer credit markets, the riskiness of a borrower can be thought of as the "good" that the lender "purchases". The assessment of risk is crucial as loans involve an agreement to pay in the future. In their seminal work, (Stiglitz & Weiss, 1981) suggested that even in a competitive equilibrium, credit markets can witness rationing (i.e. given two individuals with identical risk profiles and preferences, one will receive a loan and another will not) owing to insufficient information. Given information asymmetries, lenders rely on a combination of pricing (interest rates) and rationing to maximize returns. However, higher interest rates, while covering the risk of borrower default, are also likely to result in adverse selection. That is, higher interest rates attract borrowers seeking to make risky investments with the potential for high rates of return. (Stiglitz & Weiss, 1981) further argue that the price mechanism alone might not clear loan markets because as interest rates increase to compensate for rising risk, riskier applicants are attracted. Moreover, some borrowers will

have an incentive to make riskier investments to cover the price of credit. Faced with this “moral hazard” (the relative lack of penalty for non-payment) and with the problem of adverse selection that stems from asymmetric information, lenders will ration credit.

This all suggests that with more information about the borrowers being available, the pool of borrowers should improve, the risks of defaults should be reduced, and in some circumstances, the volume of lending should increase. The study by (Padilla & Pagano, 1997) confirms these notions: when information sharing takes place among lenders default rates are lower when information sharing takes place, interest rates are predicted to decrease and the total volume of lending to increase. In line with that, (Bennardo et al., 2015) also show in their theoretical work that information sharing reduces default and interest rates. The model of (McIntosh & Wydick, 2009) decomposes the overall effect of credit information sharing into three: a screening effect, an incentive effect with lower borrower default rates, and a credit expansion effect which increases default rates from larger loans (even though the former seem to overwhelm the latter in an overall view). In another model, (Padilla & Pagano, 2000) show that the disciplinary effect on borrowers from sharing information between lending institutions reduces default and interest rates. However, they show that this depends also on the type of information that is shared.

Thus, credit reporting systems have emerged (as early as the late 19th century where the a newly founded company Dun & Bradstreet solicited information in order to systemize a borrower's "character and assets"¹) as the means of credit information sharing to reduce information asymmetry and support the efficiency of credit institutions in their lending

¹ (Kaufman, 2018), The History of the FICO® Score, <https://tinyurl.com/yc4y2aye>

making processes, and in tasks such as credit limit management, debt collection, cross-selling, risk based pricing, prevention of fraud, etc (J. Breeden et al., 2007; Hand & Henley, 1997; Thomas & Malik, 2010). These credit reporting systems are comprised by the actors, rules, procedures, standards, and technology that facilitate the flow of information relevant to credit agreement decision making. Those actors refer to specific entities: individuals, Credit Reporting Service Providers (CRSPs), data providers, authorities, regulators, and supervisors. In particular, CRSPs can be further categorized² as follows (World Bank Group, 2019):

- *Credit bureaus* that collect and provide credit information on individuals and SMEs. More often than not these entities are private corporations or owned by the lenders. The compiled information is made available on request to customers of the credit bureau for purposes of credit risk assessment, credit scoring, or other similar purposes; consumer bureau customers include banks and other financial institutions that evaluate individuals for credit.
- *Credit registries* which generally are considered public entities and their role is to support the state and competent authorities in their supervisory and policy making responsibilities.
- *Commercial credit reporting companies* which collect information on businesses, including sole proprietorships, partnerships, and corporations. The compiled information is made available on request to customers of the

² It shall be noted here that this distinction is indicative and is not a strict taxonomy

commercial reporting company for the purposes of credit risk assessment, credit scoring, or other similar purposes, such as the extension of trade credit. Commercial credit reporting company customers include banks and other financial institutions that evaluate businesses for trade credit or insurance for business purposes

For the purpose of this thesis we shall collectively refer to all CRSPs as “credit bureaus”. One of the principal tools of credit bureaus is *credit scoring*. Credit scoring can be defined as|:

- “[credit scoring is] the term used to describe formal statistical methods used for classifying applicants for credit into ‘good’ and ‘bad’ risk classes. (Hand & Henley, 1997)
- "the use of statistical models to transform relevant data into numerical measures that guide credit decisions" (R. Anderson, 2007).
- "the set of predictive models and their underlying techniques that aid financial institutions in the granting of credits. These techniques decide who will get credit, how much credit they should get, and what further strategies will enhance the profitability of the borrowers to the lenders. Credit scoring techniques assess the risk in lending to a particular client. They do not identify “good” or “bad” (negative behaviour is expected, e.g. default) applications on an individual basis, but they forecast probability, that an applicant with any given score will be “good” or “bad”." (Rezac & Rezac, 2011)

- "A credit score is a model-based estimate of the probability that a borrower will show some undesirable behavior in the future... for example, lenders employ predictive models, called scorecards, to estimate how likely an applicant is to default (probability of default)" (Lessmann et al., 2015)

The scientific background to modern credit scoring was laid down by the pioneering work of Ronald. A. Fisher (Fisher, 1936) and it was (Durand, 1941) a little later who recognized that the same approach could be used to distinguish between good and bad loans. Nevertheless, the automated and thus widespread application of credit scoring did not take place until the 1980's, when computing power to perform sophisticated calculations became affordable and FICO developed its first scorecard using statistical methods³. For the next decades despite some methodological advances in the academic research, such as usage of artificial neural networks, SVMs, self-organizing maps, MARS (multivariate adaptive regression splines) (see indicatively, (Boyacioglu et al., 2009; F.-L. Chen & Li, 2010; Ping & Yongheng, 2011; Sarlija et al., 2006; C. F. Tsai & Wu, 2008; West, 2000; P. Yao, 2009) etc.) the field (and the practice of credit scoring) remained largely unchanged; credit scoring has relied on linear statistical methods (mainly logistic regression) and a limited number of fixed variables to calculate a borrower's credit score. This changed after 2010 where the proliferation of "big data" combined with the successful application of more sophisticated Machine Learning (ML) methods such as "deep learning" (referring to multi-layered neural networks) (Hinton & Salakhutdinov, 2006), Deep Neural

³ (Kaufman, 2018), The History of the FICO® Score, <https://tinyurl.com/yc4y2aye>

Networks (LeCun et al., 2015) and similar advancements revolutionized the field of credit scoring (among others).

Despite all the radical advances, credit scoring still faces many methodological and practical challenges such as:

- Lack of adequate, real-world and large-scale credit related data. Small datasets have been noted in the literature that may introduce unwanted artifacts and the models built upon them do not scale up when put into practice (Jamain & Hand, 2009; Perlich et al., 2003).
- All predictive models suffer 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; credit scoring is no exception (Adams et al., 2010; Bifet et al., 2011; Gama et al., 2004, 2014; Klinkenberg, 2004; Žliobaitė, 2009; Žliobaitė et al., 2016). To tackle this problem in practical terms, credit bureaus implement continuous monitoring cycles and periodic re-calibration or re-development of their models (R. Anderson, 2007; Jung et al., 2015; Siddiqi, 2005).
- Development of behavioral 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. relationships

- (especially on untransformed data) and increase the performance of generalized linear models (R. Anderson, 2019), which are even today the "golden standard" in the credit scoring industry (although to a far lesser extent than in past decades, due to the above mentioned proliferation of ML methods).
- This proliferation, on the other hand, besides (expected) performance improvements (Alonso & Carbó, 2020) introduced issues such as transparency, bias and fairness (Bussmann et al., 2020; Gilpin et al., 2018; Guidotti et al., 2018; Hardt et al., 2016a; Suresh & Gutttag, 2019; Zafar et al., 2017) which in the context of credit scoring have received special attention (N. Aggarwal, 2021; Hurlin et al., 2021; Kozodoi et al., 2022) especially in light of the statutory and regulatory constraints (cf. GDPR, EU AI Act: COM/2021/206 final).
 - From a purely methodological standpoint, besides the advances in developing credit scoring models with novel methods, there are also advancements that received little attention in the literature such as a) use of novel performance measures and b) statistical comparison between classifiers (Lessmann et al., 2015).
 - Specifically, regarding point (a), most studies rely on a single performance measure or measures such 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 (Hand & Anagnostopoulos, 2013). A coherent alternative namely the H-measure (Hand, 2009; Hand & Anagnostopoulos, 2013, 2021) has been proposed, which to the author's knowledge is not frequently used.

- Regarding point (b) statistical hypothesis testing is often neglected or employed inappropriately. Common mistakes include using parametric tests (e.g., the t-test) or performing multiple comparisons without controlling the family-wise error level. The approaches are inappropriate because the assumptions of parametric tests are violated in classifier comparisons (Demsar, 2006). Similarly, pairwise comparisons without p-value adjustment increase the actual probability of Type-I errors beyond the stated level of α (e.g., García et al., 2010).

1.1 Significance of the study

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 (Hand, 2009) and

comparing classifiers using use the Friedman's aligned ranks with post-hoc Nemenyi test (Demsar, 2006).

We thus explore the following 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 affects model performance?

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 in total more than 3,520,000 records and 125 variables.
- 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.

- We focus on the performance aspect and we compare statistical classification models versus well-advertised machine learning methods using appropriate performance measures and statistical comparison testing.

The structure of this thesis is as follows:

- Section 2 provides the related theoretical background and reviews the corresponding research literature emphasizing in the areas of concept drift, advancements and challenges in credit scoring, adaptive and local classification.
- Section 3 describes the overall experimental setup and formulates the problem.
- Section 4 provides the empirical results and
- section 5 concludes with discussion of these results and possible directions of future work.

2 Background and Related Theoretical Framework

2.1 Credit Scoring

We can roughly summarize the different kinds of credit scoring as follows⁴ based on the objective of the modeling and the data availability/usage by the predictive model (Bijak & Thomas, 2012; Paleologo et al., 2010; Phua et al., 2010):

- **Application scoring:** it refers to the assessment of the credit worthiness for new applicants. It quantifies the risks, associated with credit requests, by evaluating the social, demographic, financial, and other data collected at the time of the application. Application scoring models quantify the probability of default, by taking characteristics found in loan applications e.g. demographic attributes (such as age and family status), salary etc. This is historically the first type of credit scoring developed and by far the most researched and widely applied.
- **Behavioral scoring:** it involves principles that are similar to application scoring, with the difference that it refers to existing customers. As a consequence, the analyst already has evidence of the borrower's behavior

⁴ We shall note here that there is an ever expanding body of research in credit scoring –and especially behavioral scoring- to support decisions in areas such as marketing, through the use of propensity scores (Bijak, 2011; Thomas, 2003; Thomas et al., 2005); there are response models (will the consumer respond to marketing offers), usage models (will the consumer use a credit line) and attrition models (will a customer continue with the lender). A recent trend is also profit scoring, that is the use of scorecards to maximize profit (Andreeva et al., 2007; J. N. Crook et al., 2007; Finlay, 2010).

with the lender. Behavioral scoring models analyze the consumer's behavioral patterns to support dynamic portfolio management processes. The extra information in behavioral models is data based on the credit lines' repayment performance. We shall note here that the distinction between behavioral and application scoring is not clear-cut in the sense that if an existing customer applies for a new credit line all available information (behavior and application data) will be used.

- **Collection scoring:** collection scoring is used to divide customers with different levels of insolvency into groups, separating those who require more decisive actions from those who don't need to be attended to immediately. These models are distinguished according to the degree of delinquency (e.g. early, middle, late recovery) and allow a better management of delinquent customers, from the first signs of delinquency (30–60 days) to subsequent phases and debt write-off
- **Fraud detection:** fraud scoring ranks the applicants according to the relative likelihood that a credit application may be fraudulent.

In terms of dependent variable there are credit scoring models that estimate probability of default (PD), the exposure at default (EAD), and the loss given default (LGD) in accordance with Basel II Capital Accord requires financial institutions to estimate, respectively. Although PD models are especially well researched and continue to attract much interest, EAD and LGD models have become as well a popular research topic (e.g., (Bag & Jacobs, 2012; Bellotti & Crook, 2012; Calabrese, 2014; Gürtler et al., 2018;

Kaposty et al., 2017; K. Li et al., 2021; Loterman et al., 2012; Tobback et al., 2014; E. N. Tong et al., 2016, 2016; Yang & Tkachenko, 2012; X. Yao et al., 2015).

The techniques utilized in building credit scoring models rely mostly on classification methods and can be roughly categorized into groups such as (Abdou & Pointon, 2011; Lessmann et al., 2015; L. Yu, Wang, & Lai, 2008):

1. Statistical models: Logistic, probit or linear regression, linear discriminant analysis (LDA), classification trees, k-nearest neighbor etc.
2. Survival analysis: The latter facilitates estimating not only whether but also when a customer defaults (E. N. C. Tong et al., 2012). In addition, a special type of survival model called mixture cure model facilitates predicting multiple events of interest; for example default and early repayment (Dirick et al., 2015; F. Liu et al., 2015).
3. Mathematical programming methods: linear programming, integer programming, etc.
4. Artificial intelligence approaches (also referred as machine learning or data mining or soft computing techniques⁵): These include classic techniques like artificial neural networks, and support vector machines, as well as expert-based ones like genetic algorithms, fuzzy logic, rough sets, etc. However, recently more sophisticated ML methodologies such as Deep Neural Networks (DNN), Gradient Boosting Machines (GBM) and Random Forests (RF) came into play significantly impacting credit scoring research as well as practice (Bhatore et al., 2020; Dastile et al., 2020). We will address these developments specifically in the next section.

⁵ For the rest of the thesis we will refer collectively to these methods as “Machine Learning” or ML.

5. Hybrid and Ensemble methods: These methods include hybrid approaches (a hybridization approach is based on combining two or more different machine learning techniques, but only one single predictor is applied (Verikas et al., 2010)). For example, a hybrid classification model can be composed of one unsupervised learner (clustering method) to pre-process the training data and one supervised learner (classifier) to learn the clustering result or vice versa (C.-F. Tsai & Chen, 2010). Similar to hybrid approaches, an ensemble of classifiers uses more than one predictors but (unlike hybrid methods) the final prediction aggregates in some way the outputs of them.

Conventionally, the most widely applied method in the credit scoring industry was logistic regression (Thomas et al., 2005) followed by other linear methods, such as LDA. This preference is not without a good reason since linear models provide in practice a very good compromise between classification accuracy (compared with soft computing methods) and simplicity and interpretability (L. Yu, Wang, Lai, et al., 2008). Especially financial institutions are more reluctant to adopt less intuitive, "black box" approaches (Sousa et al., 2013) since their legislative and operational framework imposes constraints⁶ on data availability, transparency, verifiability and interpretability of their risk evaluation methods and processes.

⁶ Just to name a few: the European Consumer Credit Directive 2008/48/EC stipulates among other that an applicant has the right to be comprehensively informed about the reasons of a rejection; The Basel Accords (<http://www.bis.org/publ/bcbsca.htm>) imposes specific requirements for risk evaluation that have to be accredited. See also GDPR and recent EU AI act (COM/2021/206 final)

2.1.1 Credit Scoring Formalization

Credit scoring models in general classify customers into dichotomous “good” or “bad” (non-default / default) risk classes⁷ which indicate the probability of a credit line to be repaid or not respectively. Assuming a classification train set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, $x \in \mathbb{R}^n$, $y \in \{0, 1\}$, where x_i denotes the *feature or attribute vector* for each one of the $i=1 \dots n$, prospective borrowers and y_i the corresponding class label. Thus, according to Bayesian Decision Theory (Duda et al., 2000) credit scoring refers to calculating the probability of an applicant being “good” given its feature vector x :

$$p(G|x) = \frac{p(x|G)p(G)}{p(x)}$$

Where:

$p(x|G)$ and $p(x|B)$ refer to the conditional probabilities of the risk classes (the distributions of the classes)

$p(G|x)$ refers to the posterior probabilities of classes.

$p(G)$ and $p(B)$ denote the proportion of applicants who are good or bad correspondingly (prior probabilities).

2.1.2 Recent Advances

As mentioned, credit scoring prior to 2010 mainly has relied on linear statistical methods and a limited number of fixed variables to calculate a borrower’s credit score. This approach reflects both demonstrated statistical correlation between a borrower’s credit history and their likely credit risk, as well as traditional limits on lenders’ access to non-financial, non-credit-related data about borrowers or credit data from new non-traditional or alternative lenders (e.g. “payday lenders”, “buy now pay later” schemes, peer-to-peer

⁷ Multi-class credit risk classification has not been extensively studied or applied in practice (see (Y. Chen, 2012; Hsieh et al., 2010; Tang & Qiu, 2012) for an example of multi-class SVM for credit scoring).

lending etc who do not participate in the formal credit reporting system). The massive growth in the volume of available data, and advances in ML methods starting in the mid-2000s, combined with the contraction in bank lending following the 2008 global financial crisis, gave rise to “algorithmic credit scoring” (N. Aggarwal, 2021). Algorithmic credit scoring⁸ builds on traditional credit scoring in two principal ways:

- (i) by leveraging a much larger volume and variety of data (so-called “alternative data” or “big data”)⁹ for credit scoring; and
- (ii) by using more sophisticated ML techniques to analyze these data.

Alternative data have been studied in the context of credit scoring, ranging from call-detail records (Óskarsdóttir et al., 2019), social network and media data (Gül et al., 2018; Wei et al., 2015), utility and rent data (Michael Turner et al., 2015; Turner et al., 2012; Turner & Aggarwal, 2008), to psychometric data (Djeundje et al., 2021) and digital footprints (Tobias Berg et al., 2018). The usefulness of alternative data especially for “thin-file” or “no-file”¹⁰ prospective borrowers has been firmly established so that European credit bureaus are trying to widen their databases with such data (

⁸ Here the terms “algorithmic” is not used as a connotation to a computerized execution of the credit scoring models; in that sense conventional forms of statistical credit scoring are also ‘algorithmic’.

⁹ There is no consensus on a single definition of “alternative data”; it usually refers to data that is generated by the increasing use of digital tools and information systems (ICCR, 2018). Two categories of alternative data can be identified with respect to credit scoring: (i) *structured data*, for example, rental, utility and mobile phone payment data, transactional data, data on transactions from P2P lending platforms, invoices, accounts payables etc and (ii) *unstructured data* such as e.g. digital footprints from social media and internet usage, emails, GPS data, mobile usage, psychometric data etc.

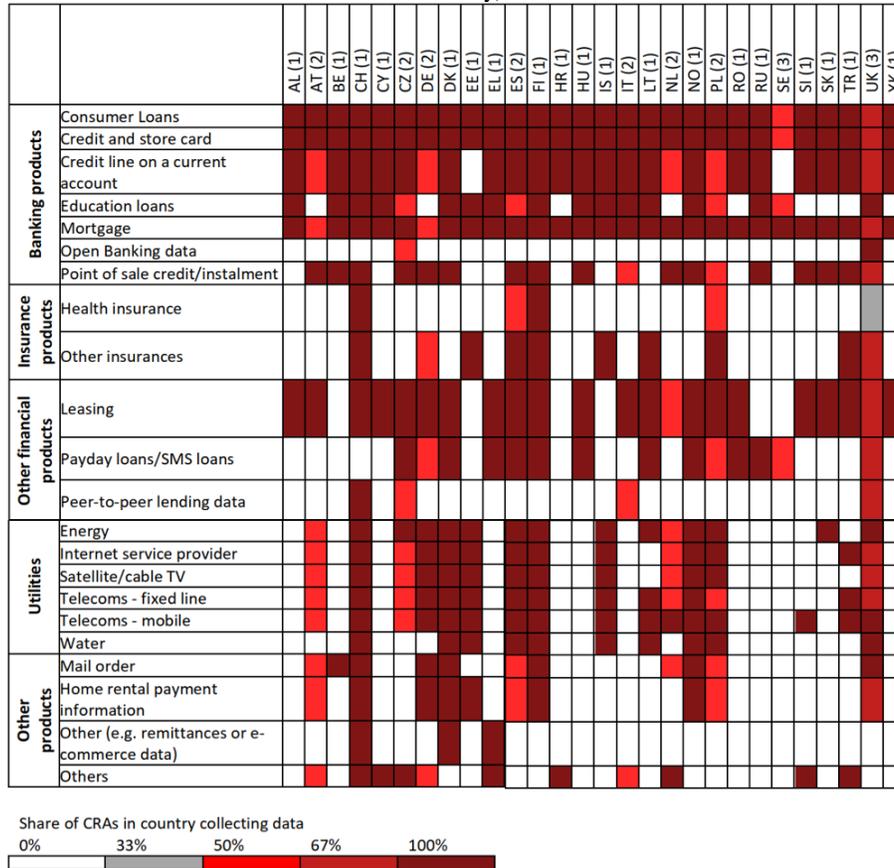
As far “big data” is concerned they are characterized typically in terms of 5 Vs: Volume, Variety, Velocity, Veracity and Value. Strictly speaking they are not identical to alternative data; however both definitions are vague enough and we use these terms interchangeably in this thesis.

¹⁰ Customers with very few or non-existent traditional financial data, which are considered also “credit invisibles” since they do not meet the mainstream criteria for getting credit.

Table 1).

Table 1: Data coverage width of European Credit Registries (CRAs)

(Source: ACCIS (2020), ACCIS Membership Survey 2020¹¹, numbers above countries indicate numbers of bureaus per country)



Appendix A provides an example of traditional and alternative data used for credit scoring. A special mention shall be given to transactional data¹¹ where their application in credit score (since they are closely related to payments) has been well studied (Hibbeln et al., 2019; Tobbach & Martens, 2019; Torrent et al., 2020). Table 2 highlights some

¹¹ The Revised Payments Services Directive (PSD2) Directive (EU) 2015/2366 PSD 2 requires banks to provide access to their customers’ payment account data to third-party providers of payment services, subject to customer consent, to enable them to offer new, differentiated services based on the use of these data. PSD2 proliferated the usage of transactional data in a multitude of applications and catalyzed open banking (Stiefmueller, 2020) and the digitization of services both in traditional banking institutions as well as it spawned an entire new “breed” of financial service providers (neo-banks, digital banks, challenger banks etc fintech companies)

indicative products from credit bureaus and fintech companies that utilize alternative data in credit scoring:

Table 2: Alternative credit scoring products
(Source: (Hurley & Adebayo, 2016) and author's analysis)

| Company & Product | Example Data Inputs |
|------------------------------|--|
| LexisNexis - RiskView | Residential stability, asset ownership, life-stage analysis, property deeds and RiskView mortgages, tax records, criminal history, employment and address history, liens and judgments, ID verification, and professional licensure |
| FICO – Expansion score | Purchase payment plans, checking accounts, property data, public records, demand deposit account records, cell and landline utility bill information, bankruptcy, liens, judgments, membership club records, debit data, and property asset information. |
| Experian – Alternative Data | Rental payment data, public record data, transactional data |
| Equifax – Decision 360 | Telco utility payments, verified employment, modeled income, verified income, spending capacity, property/asset information, scheduled monthly payments, current debt payments, debt-to-income ratio, bankruptcy scores. |
| TransUnion - CreditVision | Address history, balances on trade lines, credit limit, amounts past due, actual payment amount. |
| ZestAI | Major bureau credit reports and other variables including financial information, technology usage, and how quickly a user scrolls through terms of service |
| Kreditech | Location data (e.g., GPS), social graphing (likes, friends, locations, posts), behavioral analytics (movement and duration on a webpage), e-commerce shopping behavior, device data (apps installed, operating systems) |
| Experian Boost | Transactional data PSD2 used to pay bills and verify positive payment history |
| Lenddo/EFL | Leverages social media data (big data) and combines it with other pieces of information, including credit bureau data if available, to develop credit scores for potential borrowers |
| Earnerst | Current job, salary, education history, balances in savings or retirement accounts, online profile data (e.g., LinkedIn), and credit card information |

Combined with alternative data, advanced ML methods created a paradigm shift for the credit scoring (Addo et al., 2018; Albanesi & Vamossy, 2019; Alonso & Carbó, 2020; Barddal et al., 2020; Bequé & Lessmann, 2017; Chang et al., 2018; Dumitrescu et al., 2022; Gunnarsson et al., 2021; Hamori et al., 2018; Kvamme et al., 2018; Luo et al., 2017; Marceau et al., 2019; Petropoulos et al., 2019; Sigrist & Hirnschall, 2019; Siham et al., 2021; Sirignano et al., 2016; Sirignano & Cont, 2018; Stelzer, 2019; Tomczak & Zięba, 2015; Tripathi et al., 2021; Xia et al., 2017). Surveys conducted by supervising authorities (Bank of England, 2019; Institute of International Finance, 2019) show that credit institutions are gradually adopting more ML techniques in different areas of credit risk management, such as regulatory capital, provisions, credit scoring and monitoring. According to (Institute of International Finance, 2019) the most common use of ML in the financial industry is in the field of credit scoring. In this regard, credit institutions seem to have shifted their preferred use from regulatory purposes, such as capital calculation, stress testing and even provisions, to business-related solutions such as decisions on granting new credit, monitoring outstanding loans and refinancing non-performing exposures, and early-warning systems. In fact, the survey of (Institute of International Finance, 2019) reveals that 37% of the 60 international institutions consulted have fully operational ML models dedicated to automating credit scoring processes.

ML techniques can unleash the power of big data by parsing large, unstructured and high-dimensional datasets, to find features and patterns that are relevant to predicting a borrower's creditworthiness. Importantly, ML can more accurately capture nonlinear relationships in data, as well as reflect changes in the population and environment in order to more accurately estimate a borrower's creditworthiness—for example, by offsetting

evidence of historic payment default with more recent evidence of prompt payment, or factoring in expected payments from flexible working arrangements that are increasingly common in the ‘gig’ economy. The use of a much larger number of data points on the consumer can also reduce the risk that errors in the data will be determinative — for example, where living consumers are recorded as deceased (so-called ‘credit zombies’), or discharged debts remain on a consumer’s credit record (Hurley & Adebayo, 2016).

The following Figure 1 depicts the performance gain of a wide range of ML methods when compared to a logit model, by a comprehensive literature review conducted by (Alonso & Carbó, 2020). On the horizontal axis the reviewed papers are ranked by the authors based on their perceived algorithmic complexity. On the vertical axis the gain in predictive power (AUC) relative to the discriminatory power obtained using a Logit model on the same sample. While the sample sizes and the nature of the underlying model designs differ between studies, they all highlight that the more advanced ML techniques (e.g. random forest and deep neural networks) predict better than traditional statistical models. The predictive gains are very heterogeneous, reaching up to 20% and not behaving monotonically as we advance towards more algorithmically complex models. However, we empirically observe that with the exception of a few studies (Petropoulos et al., 2019; Sigrist & Hirschall, 2019; Sirignano & Cont, 2019) which are on the upper spectrum of performance gains or (Guégan & Hassani, 2018; Turiel & Aste, 2019) on the opposite (and can be considered as “outliers”), the performance gain reported from the rest of the papers lies within the range of 2% - 8%.

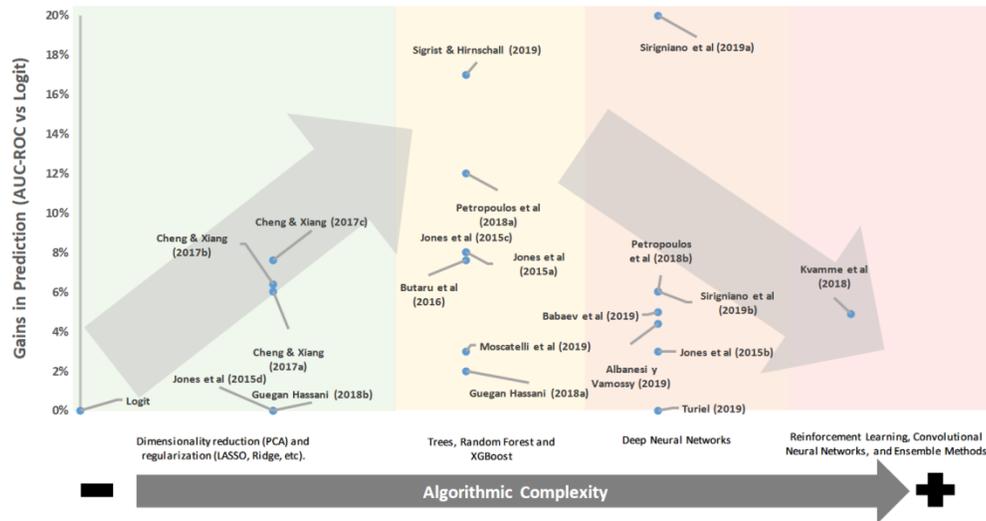


Figure 1: Performance gain of ML methods versus Logit
(Source: (Alonso & Carbó, 2020))

Despite their revolutionary potential, ML methods combined with alternative data pose some important challenges in terms of fairness, bias etc and which we will address in the following section.

2.1.3 Challenges and Issues

2.1.3.1 Challenges

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

- Lack of adequate, real-world and large-scale credit related data. Small datasets have been noted in the literature that may introduce unwanted artifacts and the models built upon them do not scale up when put into practice (Jamain & Hand, 2009; Perlich et al., 2003).
- As is the case 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. (Adams et al., 2010; Bifet et al., 2011; Gama et al., 2004, 2014; Klinkenberg, 2004; Žliobaitė, 2009; Žliobaitė et al., 2016). To tackle this problem in practical terms, credit bureaus implement continuous monitoring cycles and periodic re-calibration or re-development of their models (R. Anderson, 2007; Jung et al., 2015; Siddiqi, 2005). We will examine concept drift in detail in the following section.

- 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 predictors 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 (Avery et al., 2000).

- 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 (Thomas, 2007) or cause the relevance of predictors to change between groups (R. Anderson, 2019), b) capture non-linear relationships (especially on untransformed data) and increase the performance of generalized linear models (R. Anderson, 2019), which are even today the "golden standard" in the credit scoring industry (although to a far lesser extent than in past decades) and c) improve the prediction efficiency by treating the heterogeneous borrowers separately (Lim & Sohn, 2007). Despite the fact that there is not enough academic consensus about the effects of segmentation in scorecards' performance (Bijak & Thomas, 2012; Thomas, 2007), segmentation is a de facto approach throughout the credit scoring industry for another reason: robustness.
- From a purely methodological standpoint, besides the advances in developing credit scoring models with ML methods, there are also advancements that received little attention in the literature such as a) use of

novel performance measures and b) statistical comparison between classifiers (Lessmann et al., 2015).

- Specifically, regarding point (a), most studies rely on a single performance measure or measures such 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 (Hand & Anagnostopoulos, 2013). A coherent alternative namely the H-measure (Hand, 2009; Hand & Anagnostopoulos, 2013, 2021) has been proposed, which to the author's knowledge is not frequently used.
- Regarding point (b) statistical hypothesis testing is often neglected or employed inappropriately. Common mistakes include using parametric tests (e.g., the t-test) or performing multiple comparisons without controlling the family-wise error level. The approaches are inappropriate because the assumptions of parametric tests are violated in classifier comparisons (Demsar, 2006). Similarly, pairwise comparisons without p-value adjustment increase the actual probability of Type-I errors beyond the stated level of α (e.g., García et al., 2010).
- As mentioned, the proliferation in usage of alternative data and ML methods raise serious issues of transparency, bias and fairness (Bussmann et al., 2020; Gilpin et al., 2019; Guidotti et al., 2018; Hardt et al., 2016b; Suresh

& Guttag, 2019; Zafar et al., 2017). These issues have received special attention in the context of credit scoring (N. Aggarwal, 2021; Hurlin et al., 2021; Kozodoi et al., 2022) enhanced by the to the statutory and regulatory constraints (cf. GDPR, EU AI Act: COM/2021/206 final). Specifically (EBA, 2020) highlights the following challenges or “elements of trust” for ML as they are referred:

- **Ethics:** in line with the Ethics guidelines for trustworthy AI from the European Commission’s High-Level Expert Group on AI¹², the development, deployment and use of any AI solution should adhere to some fundamental ethical principles, which can be embedded from the start in any AI project, in a sort of ‘ethical by design’ approach.
- **Explainability and interpretability:** A model is explainable when its internal behavior can be directly understood by humans (interpretability) or when explanations (justifications) can be provided for the main factors that led to its output. The significance of explainability is greater whenever decisions have a direct impact on customers/humans and depends on the particular context and the level of automation involved. Explainability is just one element of transparency. Transparency consists in making data, features,

¹² <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>

algorithms and training methods available for external inspection and constitutes a basis for building trustworthy models.

- **Fairness and avoidance of bias:** Fairness requires that the model ensure the protection of groups against (direct or indirect) discrimination. Discrimination can be a consequence of bias in the data, when the data are not representative of the population in question.
- **Traceability and auditability:** the use of traceable solutions assists in tracking all the steps, criteria and choices throughout the process, which enables the repetition of the processes resulting in the decisions made by the model and helps to ensure the auditability of the system.
- **Data and consumer protection:** consumer rights should be respected and protected in compliance with pertaining legislation (e.g. GDPR)
- **Security:** new technology trends also bring new attack techniques exploiting security vulnerabilities that need to be addressed

2.1.3.2 Criticisms about credit scoring

Despite these challenges 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)” (Thomas et al., 2002). This doesn’t mean

thought that the application of credit scoring in practice is without its criticisms, some of which are summarized below:

- Credit scores use any characteristic with a strong correlation with the dependent variable (i.e. PD in our case) in spite of whether a clear link with a likely repayment can be justified (i.e. they rely on association and not causality cf. (Fahner, 2012; G. Xu et al., 2020)).
- As with any predictive model, misclassification is always an issue and with it comes the possibility of indirect discrimination. As mentioned in (Abdou & Pointon, 2011) (citing (Chandler & Coffman, 1979) a credit scoring system can “reject a creditworthy applicant because he/she changes address or job”.
- “*Credit invisibles*” (Turner et al., 2006, 2009) (individuals or companies with not adequately credit history and data depth from which a credit score to be calculated) pose a serious problem for expanding financial inclusion. As mentioned in section 2.1.1 use of alternative data can widen the separable population (Michael Turner et al., 2015) and subsequently access to credit, especially to low income individuals (Turner et al., 2012; Turner & Agarwal, 2008).

2.2 Concept Drift and Adaptive Learning

Concept drift refers to changes in the socio-economic environment that cause the underlying distribution of the modeled population to change over time (Adams et al., 2010;

Bifet et al., 2011; Daumé III & Marcu, 2006; Gama et al., 2004, 2014; Klinkenberg, 2004; Tsymbal, 2004; Žliobaitė, 2009; Žliobaitė et al., 2016).

In general, *adaptive learning* refers to updating predictive models online during their operation to react to concept drifts. There can be distinguished two learning modes (Gama et al., 2014): *offline learning* and *online learning*. In *offline learning*, the whole training data must be available at the time of model training. Only when training is completed can the model be used for predicting. In contrast, *online algorithms* process data sequentially. They produce a model and put it in operation without having the complete training dataset available at the beginning. The model is continuously updated during operation as more training data arrives. Less restrictive than online algorithms are *incremental algorithms* that process input examples one by one (or batch by batch) and update the decision model after receiving each example. Typically, in incremental algorithms, for any new presentation of data, the update operation of the model is based on the previous one. *Streaming algorithms* are online algorithms for processing high-speed continuous flows of data.

(Gama et al., 2014) provided a taxonomy of adaptive learning based on four key components:

- (i) *Memory* i.e. which data are used for learning and which (old) data are discarded (*forgetting mechanism*). *Sliding windows* of either fixed or variable size, which store the most recent observations, are an example of memory mechanism.
- (ii) *Change detection* i.e. the techniques and mechanisms for explicit drift and change detection. It characterizes and quantifies concept drift by

identifying change points or small time intervals during which changes occur. However, an adaptive learner can also work without detecting drift e.g. online learning systems, without any explicit change detection mechanism, can adapt to evolving data.

- (iii) *Learning* component refers to the techniques and mechanisms for generalizing from examples and updating the predictive models from evolving data. For example *retraining* learning mode discards the current model and builds a new model from scratch using buffered data, whereas *incremental adaptation* updates the model.
- (iv) Finally, *loss estimation* is an estimation mechanism based on environment feedback. E.g. (Klinkenberg & Joachims, 2000) recognize and handle concept changes using the properties of support vector machines.

The evolution of data distributions over time in a dynamic, non-stationary environment (Tsybal, 2004; Widmer & Kubat, 1996; Žliobaitė, 2009) naturally affects also credit scoring. Specifically, when the population distributions change over time then we refer to *population drift*, a very common phenomenon in economy. Formally speaking, population drift can occur in three ways (Kelly et al., 1999; Pavlidis et al., 2012):

- (i) change of risk classes *prior probabilities* $p(G)$ and $p(B)$,
- (ii) change in the class *conditional probabilities* $p(x|G)$, $p(x|B)$ and
- (iii) change in the *posterior probabilities* $p(G|x)$, $p(B|x)$.

It's worth mentioning that changes in class priors and/or class conditional probabilities do not necessarily lead to change in posterior probabilities, in which case the

predictive decision will remain unaffected (Gama et al., 2014; Kelly et al., 1999). However, in reality we could only observe the changes in the joint probabilities: $p(x, G) = p(G|x)p(x)$ or $p(x, B) = p(B|x)p(x)$ making it hard to distinguish whether actually $p(x)$ or $p(\{G, B\}|x)$ has changed (Gao et al., 2007).

In order to handle population drift credit bureaus implement continuous monitoring cycles thus *retraining* (or *calibrating*) continuously their models (R. Anderson, 2007; Jung et al., 2015; Siddiqi, 2005). The calibration of credit scoring models or actually the lack thereof, has been mentioned in the literature as one reason (among others) for the subprime mortgage crisis of 2008 (Rona-Tas & Hiss, 2008). Specifically, FICO scores have been shown to having become a worse predictor of default between 2003 to 2006 (Ashcraft & Schuermann, 2008; Demyanyk & Van Hemert, 2008) that despite the rapid and severe deterioration of subprime portfolio quality, corresponding scores remained fairly stable (J. Breeden, 2014). Thus static credit scoring models based on historical data may fail to accommodate the inherent cyclicity of banking business (in accordance with the economic cycles of recession and expansion) and the shift this entails to the entire loss distribution (Allen & Saunders, 2002; Niklis et al., 2014).

2.3 Adaptive credit scoring

Tightly intertwined within *population drift*, is the degradation of the scoring models over the business cycles and the more general impact this degradation has over risk modeling (J. L. Breeden et al., 2012; J. L. Breeden & Thomas, 2008; J. N. Crook et al., 1992; Takada & Sumita, 2011). Specifically, Basel II capital accord stipulates that a rating

system that remains relatively constant through different business conditions is a “*through-the-cycle*” (TTC) rating system whilst a rating system that changes period by period is a “*point-in-time*” (PIT) rating system. Borrowers in the same risk category of a PIT rating system would share similar unstressed PDs, and borrowers in a risk category of a TTC rating system would share similar stressed PDs. Thus, the characteristics of PDs associated with each risk category are determined by the underlying rating system and the type of information used. The information needed to forecast the defaults can be aggregate information, which typically includes macroeconomic variables such as GDP growth rates, exchange rates and interest rates, and specific borrower information that includes characteristics of and relevant financial information on borrowers. A TTC score should take into consideration specific borrower characteristics plus macroeconomic conditions, (e.g., (Bonfim, 2009)), but a PIT score would be based mainly on current information on borrowers.

Thus *adaptive learning* in the context of credit scoring and risk modeling, has been approached mainly in two ways:

- (i) One approach (focusing on the *learning component* as specified by (Gama et al., 2014)) tries to incorporate macroeconomic variables in the modeling process (Bellotti & Crook, 2014; J. Breeden et al., 2007; J. Crook & Bellotti, 2010; Saha & Siddiqi, 2011; Tony Bellotti & Jonathan Crook, 2013), sometimes using two-stage models: PIT risk is captured usually through standard scorecards and then an adaptation (e.g., in the form of linear regression) captures the system risk (Papouskova & Hajek, 2019; Sousa et al., 2013, 2014). Survival analysis was often used as a methodology for

including macroeconomic variables (Bellotti & Crook, 2008), by incorporating random effects into survival models (Djeundje & Crook, 2018; Figlewski et al., 2012; Leow & Crook, 2014) or by including a time-dependency mechanism for capturing temporal phenomena in proportional hazards survival model (Im et al., 2012). Also Markov chain transition matrixes have been used to capture the dynamics of transition the PD from time $t-1$ to t (Grimshaw & Alexander, 2011; Malik & Thomas, 2012).

- (ii) Another generic approach is by using various forms of *online learning*. (Whittaker et al., 2006) proposed the application of Kalman filter to adaptively estimate parameters β as new information (i.e. from new applicants) becomes available, so that current observations are given higher weight than previous observations, which are increasingly discounted. (Anagnostopoulos et al., 2009, 2012; Pavlidis et al., 2011, 2012) propose a way of estimating logistic regression online in a temporally adaptive manner using forgetting factors, that provide a smooth means of putting less weight on older data. This approach can be regarded as a continuous analogue to sliding window methods and may be employed in conjunction with incremental updating. (Danenas & Garsva, 2012) proposed a hybrid method based on linear Support Vector Machines classification and Particle Swarm Optimization in combination with sliding windows, in order to identify general trends. (Elliott & Filinkov, 2008) use Hidden Markov Models to create a self-tuning, risk estimation model. (S. Guo et al., 2019) use a multi-stage self-adaptive classifier ensemble model. (Lim & Sohn,

2007) propose a cluster-based dynamic scoring model which predicts the borrowers' credibility by clustering the data set and setting separate classifiers for each cluster at various time points. (Sousa et al., 2016) propose a dynamic modeling framework that considers that data is processed batch-by-batch. Sequentially, at each monthly window, a new model is learned from a previous selected window, including the most recent month. (J. Sun & Li, 2011) use instance selection to develop a dynamic financial distress prediction model, by using sliding windows of different sizes.

2.4 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 (C. Aggarwal, 2014). In *instance-based* learning (also called *lazy* or *local* learning) the specific test instance (*query instance*), 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 (Aha et al., 1991; Bontempi et al., 2001, 2002; Bottou & Vapnik, 1992).

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 (Atkeson et al., 1997; Domeniconi et al., 2001, 2002; Zhang et al., 2006). Instance-based learning is related to but not quite the same as case-based reasoning (Aamodt & Plaza, 1994; Jo et al., 1997; Vukovic et al., 2012; R. Xu et al., 2016), 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 $\text{accuracy}(S) \geq \text{accuracy}(T)$, 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 (Garcia et al., 2012; Leyva et al., 2015; Olvera-López et al., 2010). 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 vs filter methods, where the selection criterion is based on the accuracy obtained by a classifier such as kNN, vs not relying on a classifier to determine the instances to be classified (Garcia et al., 2012).

However, we shall distinguish *instance selection* from *instance sampling* (de Haro-García et al., 2019), where the purpose is to formulate a suitable sampling methodology for constructing the training and test datasets from the entire available population. Instance

sampling deals in particular with issues such as sample size and sample distribution (balancing) (Ali et al., 2015; Bischl et al., 2016; Kuncheva et al., 2019; More, 2016) and has been displayed to be of significant importance for credit scoring due to the inherent *imbalance* in the credit scoring data domain (Crone & Finlay, 2012).

There are three primary components in all local classifiers (C. Aggarwal, 2014; Aha et al., 1991):

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* (Atkeson et al., 1997; Cleveland et al., 1988; Loader, 1999; Schaal & Atkeson, 1998) 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 (Nadaraya, 1964; Watson, 1964).

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 (Lessmann et al., 2015) and thus reasonably have not drawn much of the interest of the academic community nor the practitioner's for that matter. Some effort using advanced and/or hybrid methodologies such as self-organizing maps for clustering (Schwarz & Armingier, 2005), combining kNN with LDA and decision trees (F.-C. Li, 2009), clustered support vector machines (Harris, 2015), fuzzy-rough instance selection (Z. Liu & Pan, 2018), instance-based credit assessment using kernel weights (Y. Guo et al., 2016) displayed somewhat promising results, albeit bearing into consideration the issues arising from the datasets used (size, relevance, real-world applicability).

2.5 Local Regions of Competence

Ensemble methods also known as *Multiple Classifier Systems* (MCS) combine several base classifiers through a conceptual three phase process (Britto et al., 2014; Dietterich, 2000; Kuncheva, 2004, 2008):

1. Pool generation, where 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 use 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 the dynamic selection methods (Cruz et al., 2011, 2017, 2018) and there are papers that point out that this performance can be improved by better defining these regions and selecting relevant instances (Cruz et al., 2017; V. García et al., 2012; V. García, Sánchez, et al., 2019). One of the most common methodologies for defining local regions is kNNs (including its variations such as extended kNNs especially for imbalanced data, which as mentioned is of particular importance to credit scoring), but methods such as clustering (e.g. K-Means) (Kuncheva, 2000; Soares et al., 2006) can also be found in the literature.

Dynamic selection techniques in the context of credit scoring have received the attention in the literature (Abellán & Castellano, 2017; Ala'raj & Abbod, 2016a, 2016b; Feng et al., 2018; He et al., 2018; Lessmann et al., 2015). E.g. in a recent paper (Melo Junior et al., 2020) have proposed a modification to the kNN called Reduced Minority kNNs (RMkNN) which aim 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 (Bischi et al., 2016; Crone & Finlay, 2012; V. García, Marqués, et al., 2019; He et al., 2018; Marqués et al., 2012; Zhang & Liu, 2019) which 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 (Lasota et al., 2014; Masoudnia & Ebrahimpour, 2014; Titsias & Likas, 2002; L. Xu & Amari, 2009). This method is established based on a divide-and-conquer principle (Jacobs et al., 1991) where the feature space is partitioned stochastically into a number of subspaces through special employed error function and “experts” become specialized on each subspace. However, the main problem is that as base classifier is used only multilayer perceptron neural networks (Britto et al., 2014; Cruz et al., 2018). Mixture of Experts has not been extensively applied in the context of credit scoring and there are but a few studies on the subject (Liang et al., 2021; West, 2000; J.-M. Yu, 2018).

2.6 Imbalanced Classification

Imbalanced datasets occur as the number of observations in one class (referred to as a minority class) in a dataset is usually much lower than the number of observations in the other class (referred to as a majority class). There are quite a few studies and approaches in literature analyzing the impact of imbalancing in classification in general (Ali et al., 2015; Branco et al., 2016; Ganganwar, 2012; Kaur et al., 2019; Rahman & Davis, 2013; Sarmanova & Albayrak, 2013; Y. Sun et al., 2009; Q. Wang et al., 2017; S. Wang et al., 2018) and within the context of credit scoring in particular (Bischi et al., 2016; Brown & Mues, 2012; Crone & Finlay, 2012; V. García et al., 2012; He et al., 2018; Marqués et al., 2012).

For example, (Brown & Mues, 2012) showed that the random forest and gradient boosting classifiers perform very well in a credit scoring context and are able to cope

comparatively well with pronounced class imbalances in the datasets. On the other hand, when faced with a large class imbalance, the C4.5 decision tree algorithm, quadratic discriminant analysis and k-nearest neighbors perform significantly worse than the best performing classifiers. (Douzas et al., 2021; Douzas & Bacao, 2017, 2018) tackle the problem of imbalanced datasets by using a novel oversampling method, Self-Organizing Map-based Oversampling (SOMO). There are a number of over-sampling (applied on minority class) or under-sampling techniques (applied on majority class) that can be found in literature. For example, (Chawla et al., 2002) proposed Synthetic Minority Over-sampling Technique (SMOTE). The SMOTE over-samples the minority class by taking each minority class sample and creating synthetic examples (along the line segments joining any/all of the k minority class nearest neighbors). Thereafter, neighbors from the k nearest neighbors are randomly chosen, depending on the amount of over-sampling required. For instance, if the amount of over-sampling needed is 300%, only three neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated by taking the difference between the feature vector (sample) under consideration and its nearest neighbor. Thereafter this difference is multiplied by a random number between 0 and 1, and is added to the feature vector under consideration to form a synthetic feature vector. There were many applications and/or modifications of SMOTE proposed thereafter (Han et al., 2005; Qazi & Raza, 2012; Q. Wang et al., 2017). Most recently (Camacho et al., 2022; Douzas et al., 2021) proposed a modification called G-SMOTE which allows the generation of synthetic instances in a geometric region around the selected instances rather than in the line segment that joins the two selected instances. There are other techniques that neither do over-sampling nor under-sampling to deal with

class imbalance, such as wavelet data transformation and linear dependence approach. For example (Saia et al., 2018) proposed a discrete wavelet transformation to deal with imbalanced data in credit scoring. Wavelets are small waves and wavelet transform captures both the time and frequency domains. (Saia et al., 2018) approach outperformed the random forest model regardless of data distributions.

2.7 Methodological issues in classifiers' performance measures and comparisons

2.7.1 Performance measures

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, considering also the inherent imbalance of the credit scoring datasets (Japkowicz & Shah, 2011; Luque et al., 2019; Parker, 2011). (Japkowicz & Shah, 2011) defined an ontology of performance measures (Figure 2) where they categorize classifiers as follows:

- *Deterministic algorithms* output a fixed class label for each instance and hence can be better measured in terms of the zero–one loss. That is, the loss of misclassifying an example (assigning a wrong class label to the instance) is one; and zero otherwise.
- *Probabilistic classifiers*, on the other hand, issue a probability estimate on the class membership of the example for various classes. To obtain deterministic class assignments from probabilistic classifiers, typically either a maximum a posteriori (MAP) or a Bayesian estimate is considered

- *Scoring classifiers* are thresholded so as to obtain deterministic labels for test examples. In a binary classification scenario, a classifier that outputs scores on each test instance in a fixed interval $[a, b]$ can be thresholded at some point $st \in [a, b]$ such that all the examples with a score greater than st are classified as positive whereas the examples scoring less than st are labeled as negative.

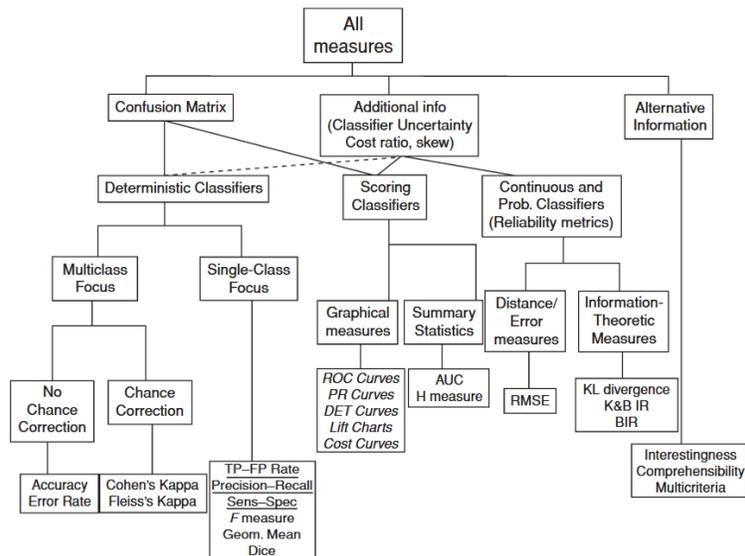


Figure 2: An ontology of performance metrics

(Source: (Japkowicz & Shah, 2011)¹³)

Specifically the credit scoring setup gives rise to methodological problems such as the accuracy paradox (Uddin, 2019; Valverde-Albacete & Peláez-Moreno, 2014) and the different misclassification cost between Type I and Type II errors (Hand, 2009). As a result, the most commonly used approach avoids accuracy as a scorecard performance metric and

¹³ KL=Kullback-Leibler, BIR=Bayesian information reward, K&B IR= Kononenko and Bratko information reward

has adopted 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 (Hand & Anagnostopoulos, 2013). A coherent alternative namely the H-measure (Anagnostopoulos et al., 2019; Hand, 2009; Hand & Anagnostopoulos, 2013, 2021) 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 (Parker, 2011). Thus in this work, we use both AUC and H-measure in accordance with above findings.

2.7.2 Comparison of Classifiers

Comparisons among classification algorithms on different datasets arise in machine learning when a new proposed algorithm is compared with the existing state of the art. (Japkowicz & Shah, 2011) identified the following cases that shall be considered upon deciding which is the appropriate approach to statistical comparison:

- The comparison of two algorithms on a single domain,
- The comparison of multiple algorithms on a single domain,
- The comparison of multiple algorithms on multiple domains.

Figure 3 depicts these possible combinations of classifiers and datasets and proposes the suitable procedure for each case. There are some points worth noting:

- In the case of two algorithms compared on multiple domains, there are proposed only non-parametric tests.

- Both ANOVA and Friedman's test, in the case of multiple algorithms compared on multiple domains, should be followed (when the null hypothesis is rejected) by a post hoc test, in order to establish where the difference was located.

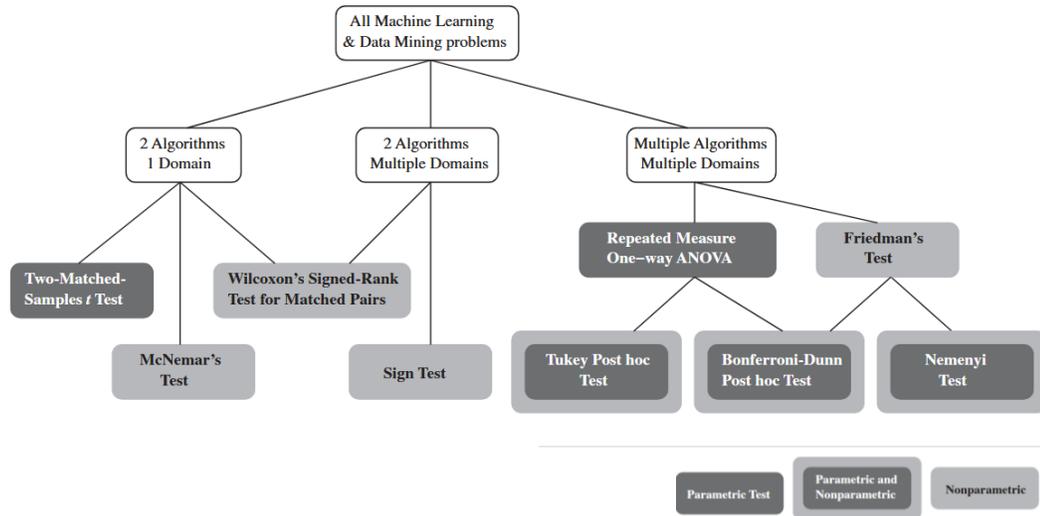


Figure 3: Statistical tests for comparing multiple classifiers

(Source: (Japkowicz & Shah, 2011)¹⁴)

Thus, from a statistical point of view, the correct way to deal with multiple hypothesis testing is by, firstly, 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

¹⁴ KL=Kullback-Leibler, BIR=Bayesian information reward, K&B IR= Kononenko and Bratko information reward

homoscedasticity in the case of ANOVA) are hardly fulfilled (Demsar, 2006; S. García et al., 2010; García & Herrera, 2008; Santafe et al., 2015). We use Friedman's aligned rank test as our omnibus test to obtain the p-values on the performance differences, testing for null-hypothesis, (i.e. that all models perform equally well, is rejected if $p < 0.05$). The chosen test is applied to the $\frac{k(k-1)}{2}$ pairwise comparisons, where k is the number of models. Due to the multiple application of the test, some p-value correction method has to be used in order to control the *familywise error rate*. This problem was tackled by (Schaffer, 1993), where there were proposed two procedures to correct the p-values:

- (i) In the first one (sometimes called Shaffer static) the particular ordering of the null hypothesis is not taken into account and only the maximum number of simultaneous hypotheses is considered.
- (ii) The second one further limits the number of possible hypotheses by considering which particular hypotheses have been rejected. This increases the power of the method, but it is computationally very expensive. Instead of this procedure, in (García & Herrera, 2008), the authors propose to use Bergmann and Hommel's method (Bergmann & Hommel, 1988)

Thus, in this work we use Friedman's Aligned Rank Test adjusted with Bergmann and Hommel's method.

Once the null-hypothesis is rejected, the Nemenyi test is performed as a post-hoc test. The Nemenyi test is used to compare classifiers pairwise, where the best performing classifier per measure is tested against all other models (Demsar, 2006).

3 Experimental Setup and Methodology

3.1 Problem Formulation

Assuming a classification train set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, $x \in \mathbb{R}^n$, $y \in \{0, 1\}$, M is a global model trained on all $\{(x_i, y_i)\}_{i=1}^n$, the local region of competence for a given test instance x (assuming its k -Nearest Neighbors) is denoted by $N_x = \{x_1, x_2, \dots, x_k\}$ and the learning set for the local classifier M_x is $\{(x_i, y_i)\}_{x_i \in N_x}$.

Specifically, for the credit scoring binary classification problem $\{x_i\}$, $i = 1, \dots, n$, 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 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 from $[T_0 - \tau', T_0]$ to $(\tau' \geq 1$ denoting months), whereas the *outcome window* refers to the period $(T_0, T_1]$ where the class label of y_i is defined (see Figure 4). 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.

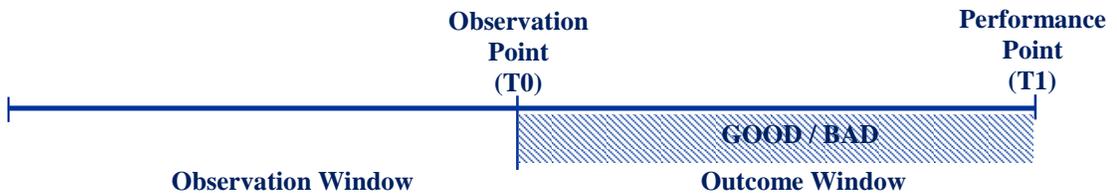


Figure 4: Observation and Outcome windows

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 (Mays, 2005; Thomas et al., 2002; Thomas & Malik, 2010).

Regarding the definition of default, Anderson (2007) 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 with the Basel II Capital Accord and used also in the new definition of default by the European Banking Authority (EBA). (Kennedy et al., 2013) have made 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 default definition outperformed longer outcome windows; for the worst status approach the degradation occurs when outcome window extends beyond 12 months.

3.2 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.

Appendix B: List of Variables provides a detailed list of variables as well as some basic descriptive statistics.

3.3 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 , all 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 for each 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 delinquency approach for each outcome window, resulting in the corresponding classes: (a) $y = 1$ is assigned to cases with worst

delinquency ≥ 90 dpd or a derogatory event during the outcome period, (b)
 $y = 0$ is assigned to all other cases

3.4 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 train 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 two 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. Specifically, Gunnarsson et al. (2021) found that XGBoost and RF outperformed Deep Belief Networks (DBN), Hamori et al. (2018) found XGB to be superior to Deep Neural Networks (DNN) an RF. Marceau et al. (2019) found that XGB performed better than DNN, and Addo et al. (2018) concluded that both XGB and RF outperform DNN.

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 for the Logistic Regression (LR). The exclusion thresholds were selected to match a practitioner's rule mentioned in the literature (Siddiqi, 2005), where a variable is removed in case of having an IV lower than 0.3 and greater than 2.5. For the ML methods we let the default (implicit) algorithm parameters on the entire feature space. However, for testing purposes we also try the same variables selected for LR (IV-based).

Finally, as it has been noted in section 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 executions¹⁵. 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.

3.4.1 Local Classification

As detailed below, for each snapshot, the k Nearest Neighbors (k-NN) algorithm is used to define its 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 5 shows the overall flow for the proposed scheme:

¹⁵ In total we executed 120 runs for local LR models (one run over all 40 snapshots for each k, where $k \in \{2000, 4000, 6000\}$ the size of kNNs).

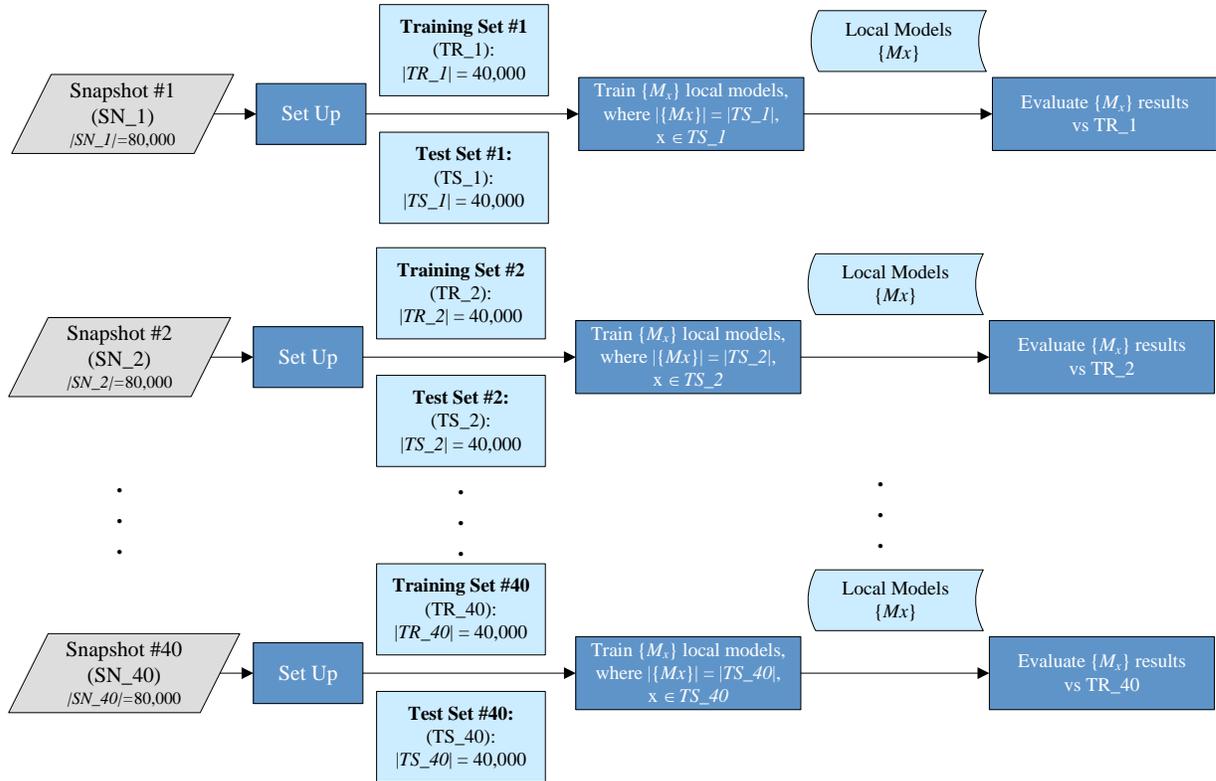


Figure 5: High-level flow for the proposed local classification scheme ($|S|$ denotes the cardinality of a set S)

The set-up 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 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 or the Mahalanobis 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 (Lewis, 1992) and mentioned in many works thereafter (R. A. Anderson, 2022; Finlay, 2010; Siddiqi, 2005). Although the subject is not extensively researched, recent academic studies pointed to the direction that larger samples can improve the performance of linear models (Crone & Finlay, 2012; Finlay, 2010) but there seems to be a plateau after 6000 goods/bads and almost no further benefit above 10000. 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 6 depicts the training phase for the proposed scheme (pre-processing refers to the flow described in section 3.4).

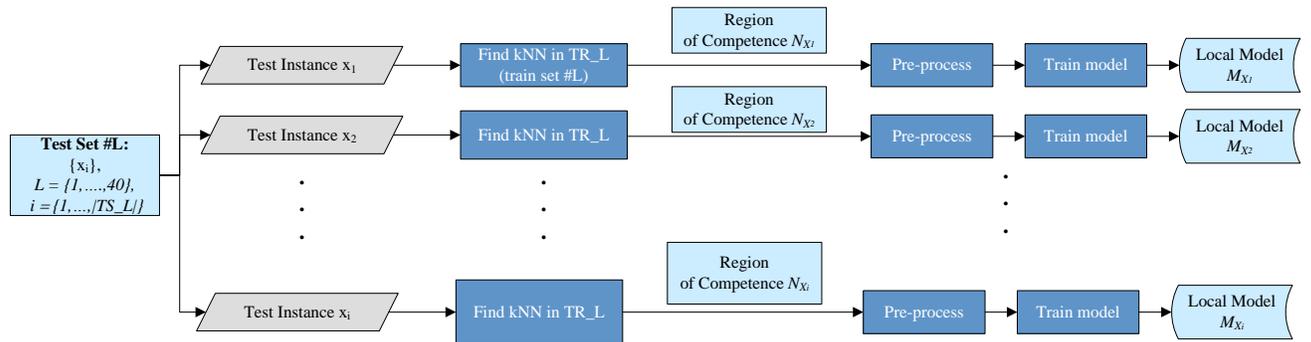


Figure 6: Training phase for the proposed local classification scheme ($|S|$ denotes the cardinality of a set S)

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} , $i=\{1,\dots,|TS_L|\}$ (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/candidate borrower and assign a GOOD or BAD class label. This is compared to the actual labels available for the test instances.

3.4.2 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 7 shows the overall flow for the global scheme.

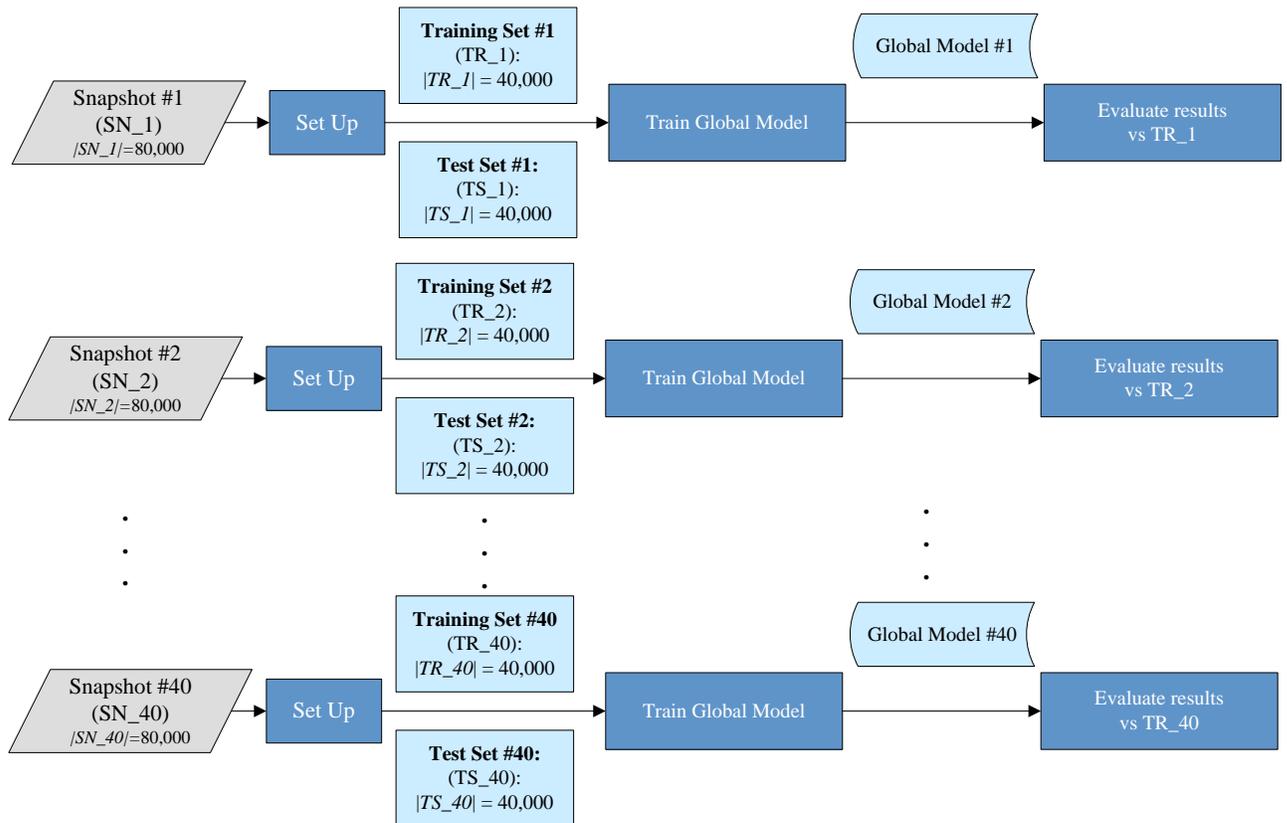


Figure 7: Global classification scheme ($|S|$ denotes the cardinality of a set S)

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

redevelopment is a process applied in practice by all commercial credit scoring models). The performance of global models over all snapshots degrades significantly in case of training only once for the initial snapshot data (see section 4.1).

4 Empirical Results

In the empirical results we use the following notation:

| | |
|------------|---|
| LR | = Logistic Regression |
| XGB | = xgboost |
| RF | = Random Forest |
| G | = Global Model |
| L | = Local model |
| 2k, 4k, 6k | = 2000, 4000, 6000, count of k for kNNs |
| euc | = Euclidian distance |
| mah | = Mahalanobis distance |
| IV | = feature selection based on Information Value (IV) process |
| FS | = implicit feature selection for ML models |
| n | = no retraining (training takes place only at 2099q1) |

4.1 Global classifiers and Population drift

We first examined the impact of population drift by training the global models at the beginning of our data period (2009q1) and compared their performance when we retrained them every 2 years (i.e. 2009q1, 2013q1, 2015q1, 2017q1), as mentioned in section 3.4.2. Table 3 summarizes the results over all snapshots. Detailed results are provided in Appendix D: Detailed Results in Table A-9 (AUC) and Table A-10 (H-Measure).

Table 3: Performance measures of global classifiers (no retrain=shaded rows) over all snapshots with different feature selection mechanisms

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain)

| Model | Mean AUC | Standard Deviation AUC | Mean H-Measure | Standard Deviation H-Measure |
|------------|----------|------------------------|----------------|------------------------------|
| LR_G_n_IV | 0.821 | 0.036 | 0.464 | 0.015 |
| LR_G_IV | 0.873 | 0.016 | 0.499 | 0.034 |
| XGB_G_n_FS | 0.899 | 0.013 | 0.577 | 0.014 |
| XGB_G_FS | 0.931 | 0.014 | 0.643 | 0.012 |
| XGB_G_IV | 0.928 | 0.012 | 0.635 | 0.012 |
| RF_G_n_FS | 0.918 | 0.011 | 0.623 | 0.014 |
| RF_G_FS | 0.933 | 0.012 | 0.659 | 0.012 |
| RF_G_IV | 0.930 | 0.012 | 0.648 | 0.012 |

Figure 8 and Figure 9 visualize the detailed performance over all snapshots for all model above results in violin plots¹⁶.

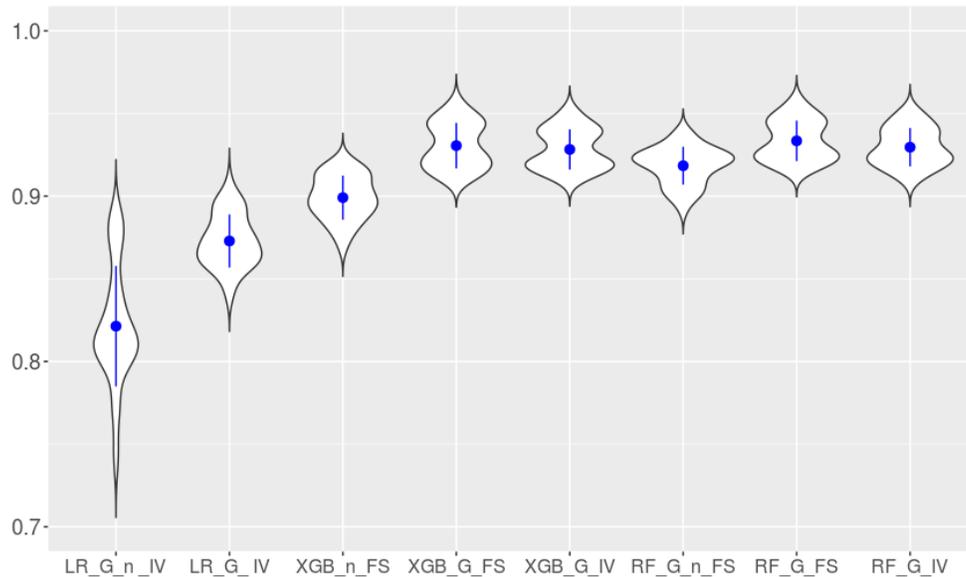


Figure 8: AUC of global classifiers (y-axis not starting from zero)

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)

¹⁶ The violin plots in this thesis display the *kernel density plot*, along with the *mean value* of the distribution (a blue dot) and *one standard deviation* above and below the mean, as a blue line.

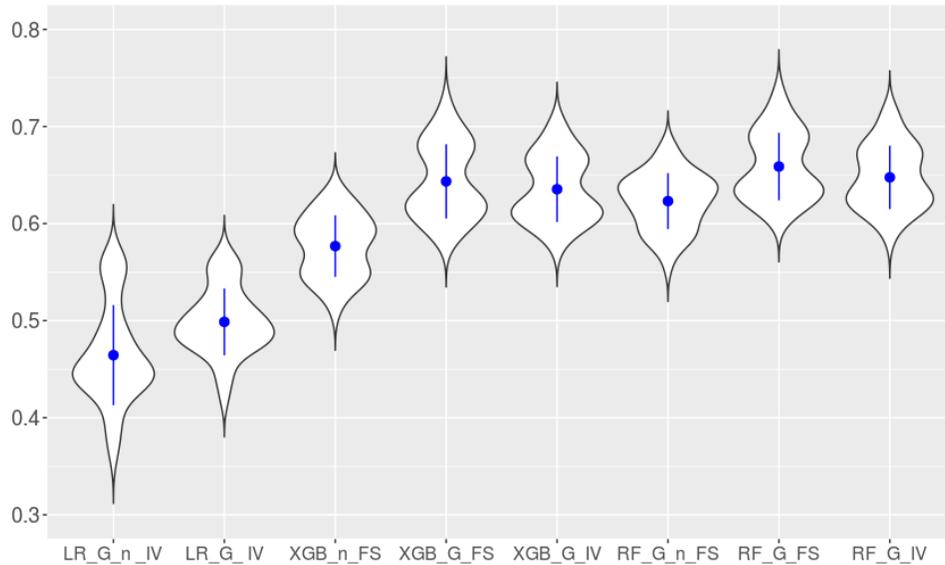


Figure 9: *H-Measure* of global classifiers (y-axis not starting from zero)

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)

From a first analysis our initial conclusions confirm the corresponding findings in literature:

- **Population drift affects model performance;** this is solidly confirmed across all models. Retraining (expectedly) benefits model performance in all cases. All models with retraining perform better than the corresponding ones without retraining.
- **XGB and RF outperform LG.** Specifically referring only to the retrained models, in terms of AUC the performance difference is 6.6% and 6.9% for XGB and RF over LG correspondingly (which is within the average range observed in other studies; see section 2.1.2) and for H-Measure the corresponding differences are 29.0% and 32.1%. We will elaborate further upon this finding when examining also local classification.

- In ML models embedded feature selection seem to outperform IV-based feature selection:** For ML models (XGB and RF) allowing them the freedom to “work” with all feature space, gives them an apparent edge over constraining them to the same set of variables (through the IV criterion) that were chosen for the LR global models.

As discussed in section 2.7.2 to test for statistical significance of these differences (i.e. comparing of multiple methods on multiple data sets as noted in Demsar (2006)) we use Friedman’s Aligned Rank Test (García et al., 2010) to obtain the p-values on the performance differences and correcting them using Bergmann and Hommel procedure (Garcia & Herrera, 2008). Figure 10 and Figure 11 visualize the corrected p-value matrices (a=0.05) for AUC and H-Measure based performances correspondingly.

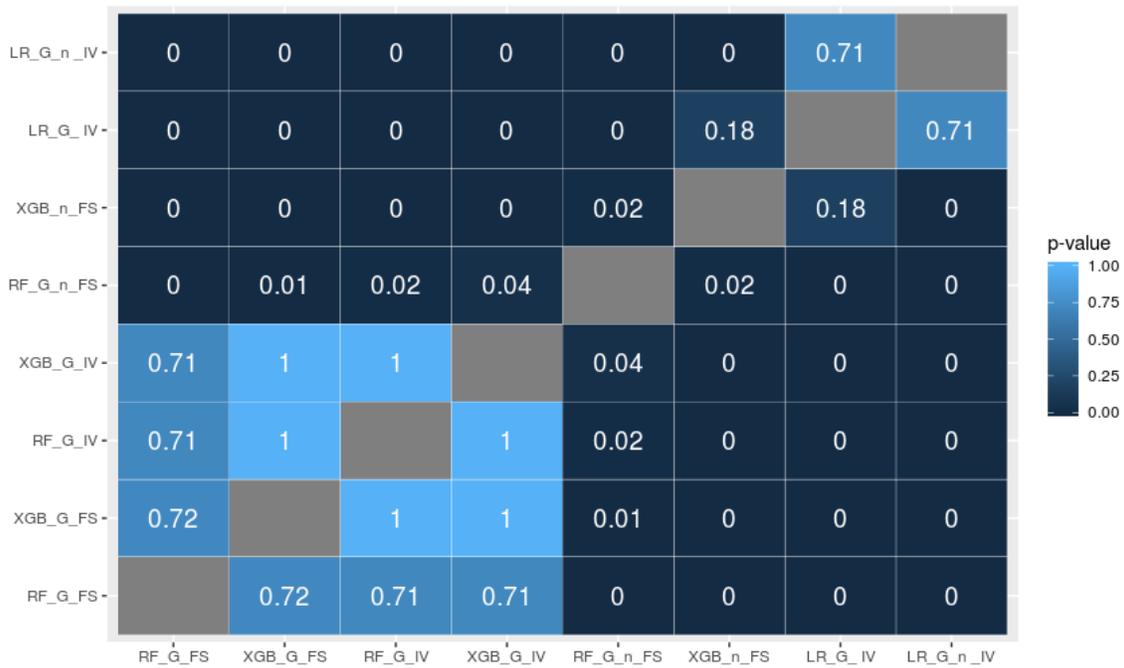


Figure 10: AUC based statistical differences of global classifiers (p-value matrix) (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain,)

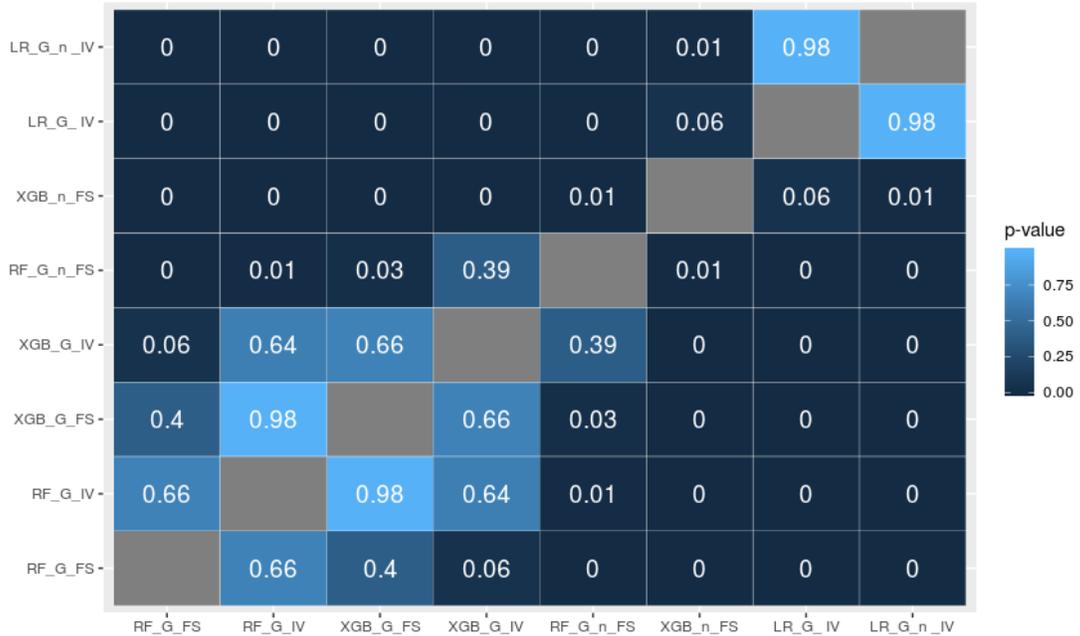


Figure 11: **H-measure** statistical differences of global classifiers (*p*-value matrix)
 (LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)

It becomes evident that RF and XGB global models with retraining are statistically similar and the differences in feature selection methods (embedded FS vs IV) for these models are not statistically significant, although FS ranks better than IV. To better depict the results we draw a ranking graph (Figure 12) where the models we compare are the nodes and two nodes are linked if the null hypothesis of being equal cannot be rejected. Within each node the average rank of the model is printed. The green node indicates the model with the highest relative rank. As it is evidenced:

- RF and XGB outperform LR in all cases.
- Retraining always outranks models trained once at the beginning of the examined period

- Feature selection method (FS vs IV) ranks always better intra-model (i.e. RF_G_FS is better ranked than RF_G_IV and XGB_G_FS is better than XGB_G_IV).
- As far as the comparison between XGB and RF is concerned, RF seem to fare better but not statistically significant.

We will elaborate on all these themes as we move in the local classifiers analysis of results.

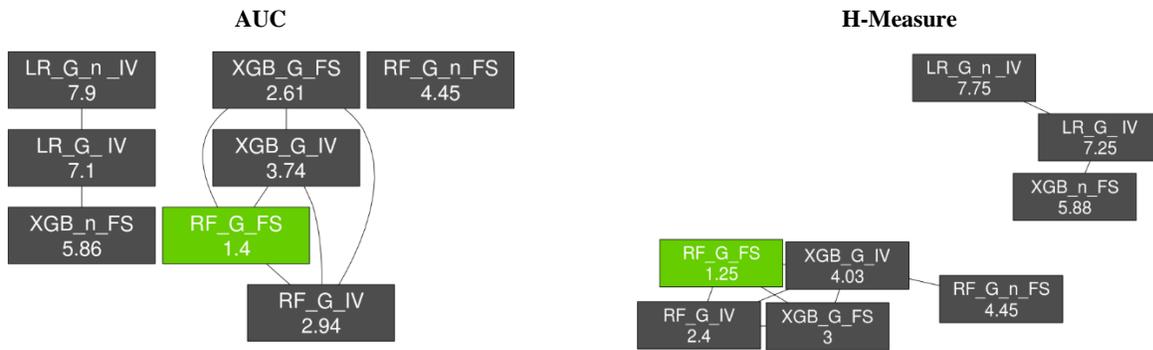


Figure 12: Graph of rankings for global models

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)

Visualizing the timeseries of performance measures (Figure 13 and Figure 14)¹⁷, we observe that additional to the apparent superiority of the retrained models over their “static” ones (in the sense of no retraining), as LR degrades quite significantly over time (-15.71% drop in AUC between 2009q1 and 2018q4 and -30.81% drop in H-Measure correspondingly), whereas XGB and RF keep their corresponding performance. Population

¹⁷ We kept only the “*_FS” models for XGB and RF (i.e. the ones fed the entire feature space, not with an IV-based feature selection

drift is also showcased as it becomes evident after the first retraining point (2011q1) where the performance deviation grows thereafter.

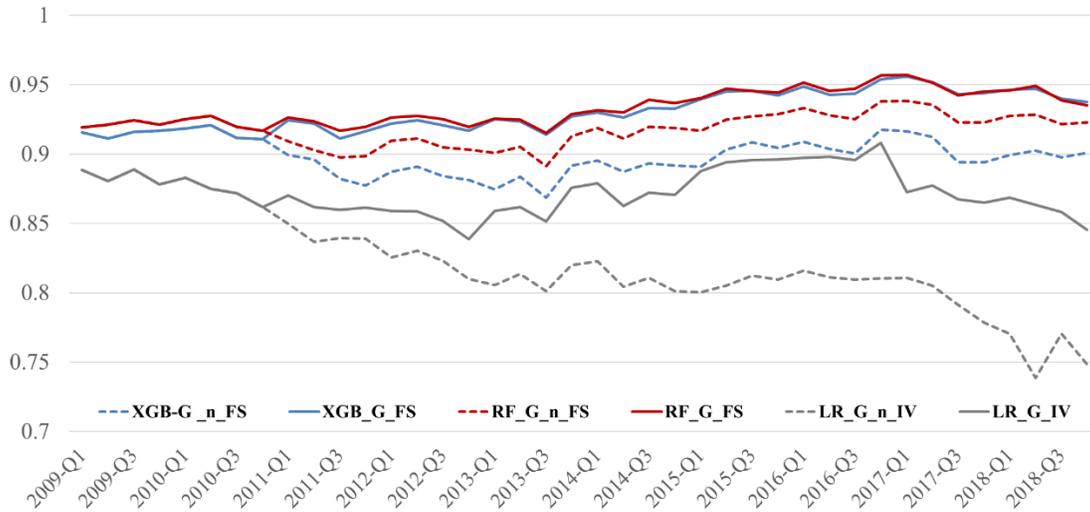


Figure 13: *AUC degradation of global classifiers with and without retraining (y-axis not starting from 0)*

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, dashed lines=no retrain)

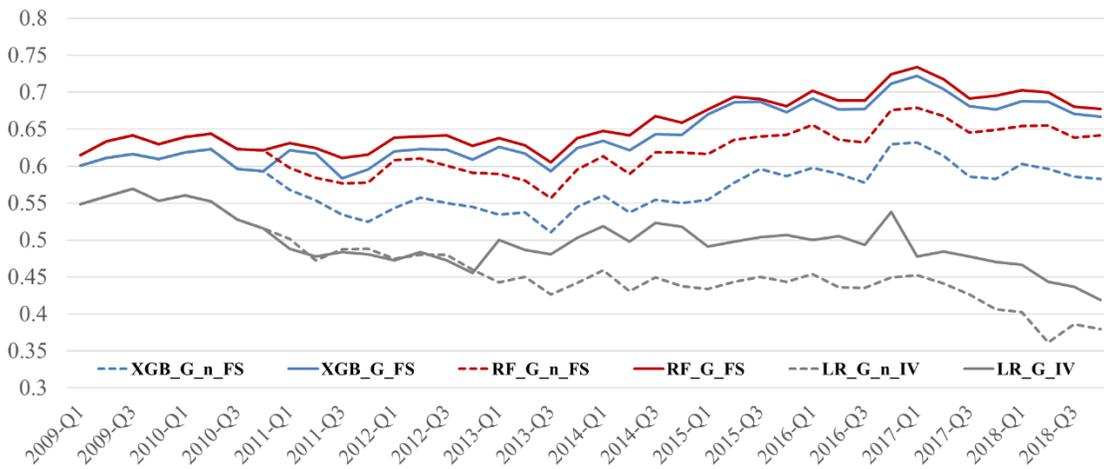


Figure 14: *H-Measure degradation of global classifiers with and without retraining (y-axis not starting from 0)*

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, dashed lines=no retrain)

Thus for the rest of the thesis we will solely use the retrained global models as our benchmark for comparing global and local models and for XGB and RF the FS option for

feature selection. For brevity in the following sections we will omit the feature selection procedure from the labels of the models (e.g. XGB_G_FS will be shorthand to XGB_G)

4.2 Local classification

4.2.1 Euclidean vs Mahalanobis Distance and size of k (kNN)

For tackling the hypothesis regarding the superiority of local models over their global counterparts, we started by examining whether the choice of distance metric and the size of the local region impacts the classification performance. Since we are using LR as a baseline we tried different kNN sizes ($k=\{2000,4000,6000\}$) for Euclidean as well the Mahalanobis distance metric for this classifier. Table 4 summarizes the performance (AUC, H-Measure) results and Table A-11 and Table A-12 in Appendix D provide the detailed results over all snapshots. Figure 15 and Figure 16 depict graphically these results.

Table 4: Performance measures of LR_L using different distance metrics and local region sizes
(LR=Logistic Regression, 2k, 4k, 6k=k in kNN, euc=Euclidion dist. mah=Mahalanobis)

| Model | Mean AUC | Standard Deviation AUC | Mean H-Measure | Standard Deviation H- Measure |
|--------------|---------------------|-----------------------------------|---------------------------|--|
| LR_L_2k_euc | 0.926 | 0.009 | 0.636 | 0.028 |
| LR_L_2k_mah | 0.891 | 0.013 | 0.559 | 0.021 |
| LR_L_4k_euc | 0.926 | 0.012 | 0.630 | 0.030 |
| LR_L_4k_mah | 0.905 | 0.008 | 0.585 | 0.021 |
| LR_L_6k_euc | 0.926 | 0.012 | 0.627 | 0.031 |
| LR_L_6k_mah | 0.911 | 0.009 | 0.595 | 0.023 |

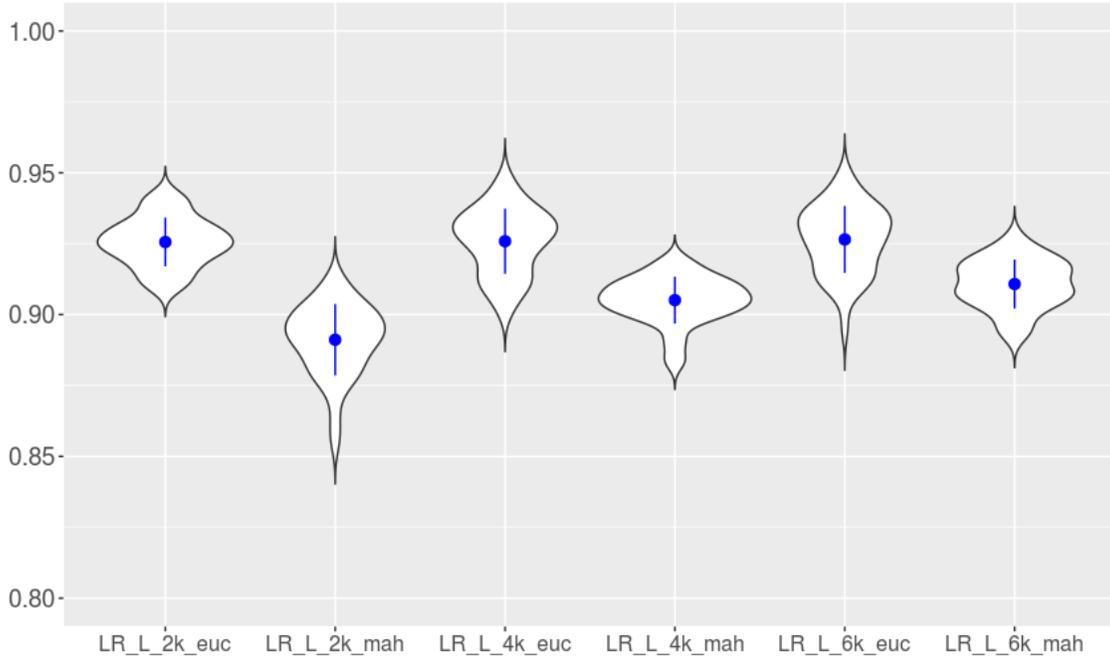


Figure 15: AUC of LR_L using different distance metrics and local region sizes (y-axis not starting from zero)

(LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidian dist. maha=Mahalanobis)

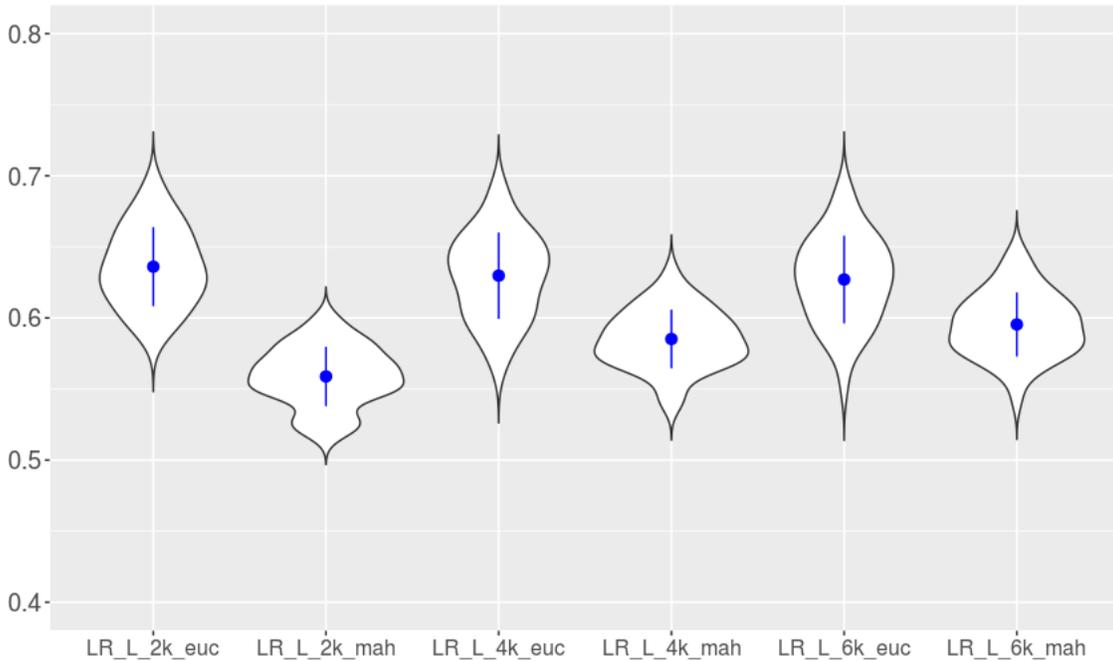


Figure 16: H-Measure of LR_L using different distance metrics and local region sizes (y-axis not starting from zero)

(LR=Logistic Regression,2k,4k,6k=k in kNN, euc=Euclidian dist. maha=Mahalanobis)

As evidenced the choice of k does not have a significant impact performance of logistic regression, when *Euclidian distance* is used as distance metric for the choice of

kNNs. 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) see Figure 17, whereas the opposite holds when using AUC as performance measure (mean=0.9256, 0.9259, 0.9265 for corresponding k 's)- see Figure 18.

However, there is an obvious a statistically significant difference between Euclidean distance and Mahalanobis distance in the choice of local regions, with the first clearly outperforming the second one. The p-value matrixes (Figure 17, Figure 18) indicate two distinct “groups” (Euclidean - Mahalanobis); whereas the null hypothesis can't be rejected intra-group, we safely can reject the inter-group null hypothesis.

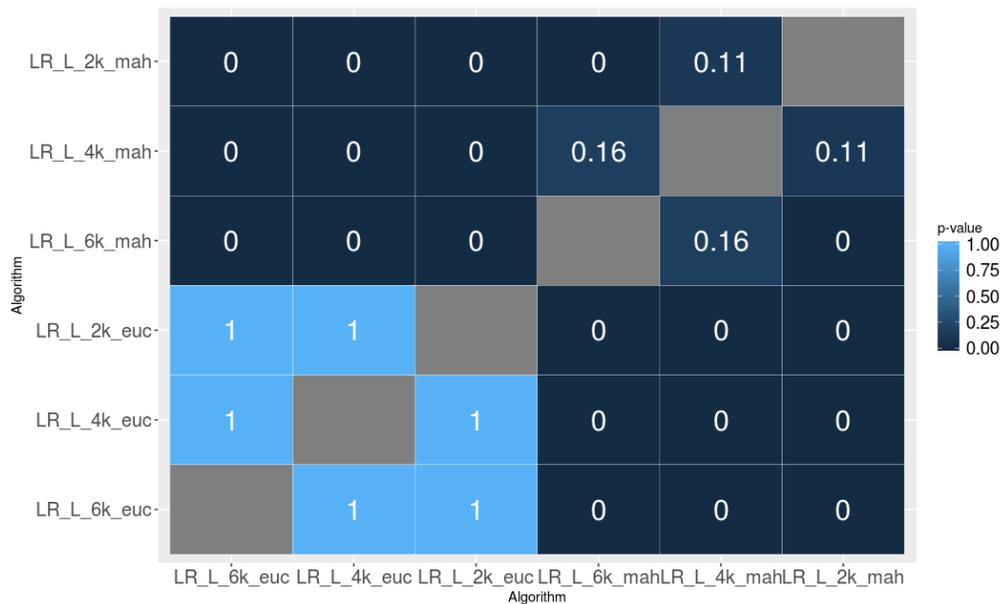


Figure 17: AUC based statistical differences of LR-L using different distance metrics and local region sizes (p-value matrix)

(LR=Logistic Regression, 2k, 4k, 6k=k in kNN, euc=Euclidion dist. maha=Mahalanobis)

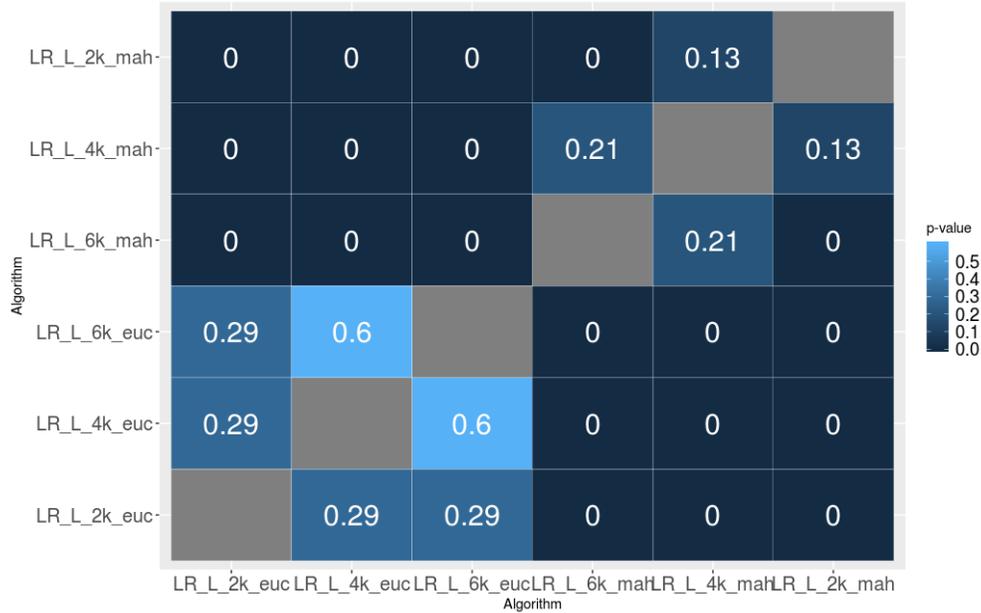


Figure 18: **H-Measure** based statistical differences of LR-L using different distance metrics and local region sizes (*p*-value matrix)

(LR=Logistic Regression, 2k, 4k, 6k=k in kNN, euc=Euclidian dist. maha=Mahalanobis)

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 and we will use only Euclidian distance.

4.2.2 Local vs Global Classifiers

Summing up our findings this far, we compared the following classifiers:

LR-L_2k = Local LR using 2000 kNNs with Euclidean distance (and IV feature selection)

XGB-L_2k = Local XBG on the same local regions with LR (no IV features)

RF-L_2k = Local RF on the same local regions with LR (no IV features)

- LR-G = Global LR with retraining every 2 years (and IV feature selection)
- XGB-G = Global XGB with retraining every 2 years (no IV features)
- RF-G = Global RF with retraining every 2 years (no IV features)

Comparing visually the performance timeseries of the local classifiers with their corresponding global ones, we get a mixed picture (see Table A-13 and Table A-14 in Appendix D: Detailed Results): whereas local LR models characteristically outperform their global counterparts, for XGB and RF the differences between global and local classifiers do not appear to be significant (Figure 19).

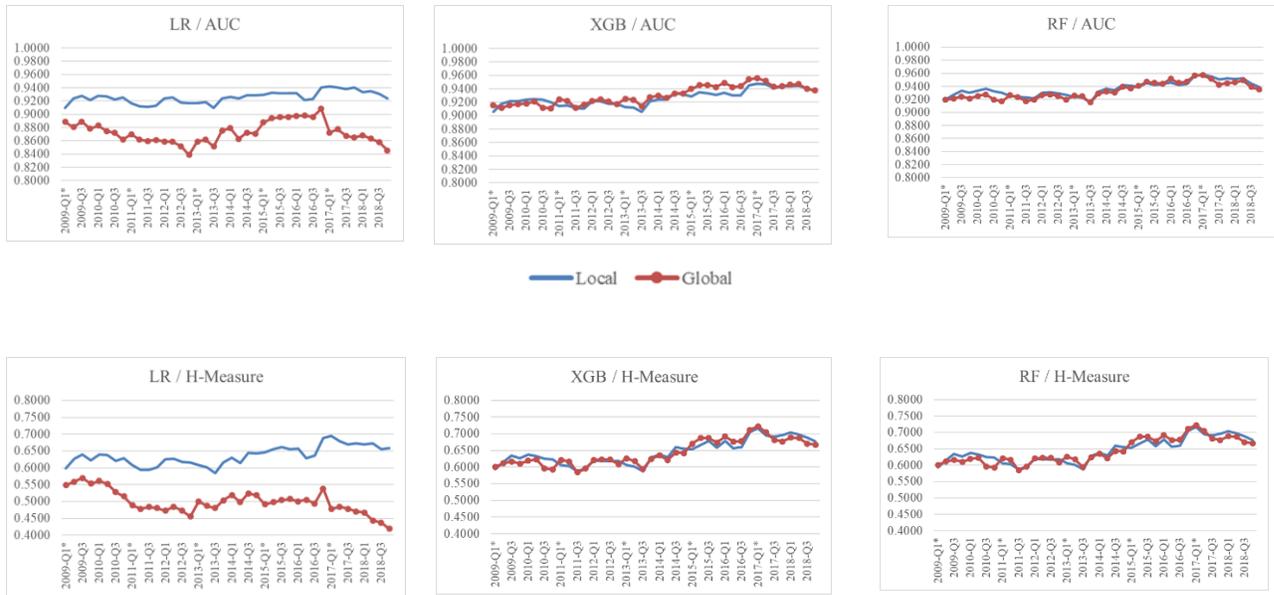


Figure 19: Pairwise timeline comparison between local/global classifiers sizes (y-axis not starting from zero) (different y-axis scales, LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, solid blue line denotes local classifier, red line with markers global classifier)

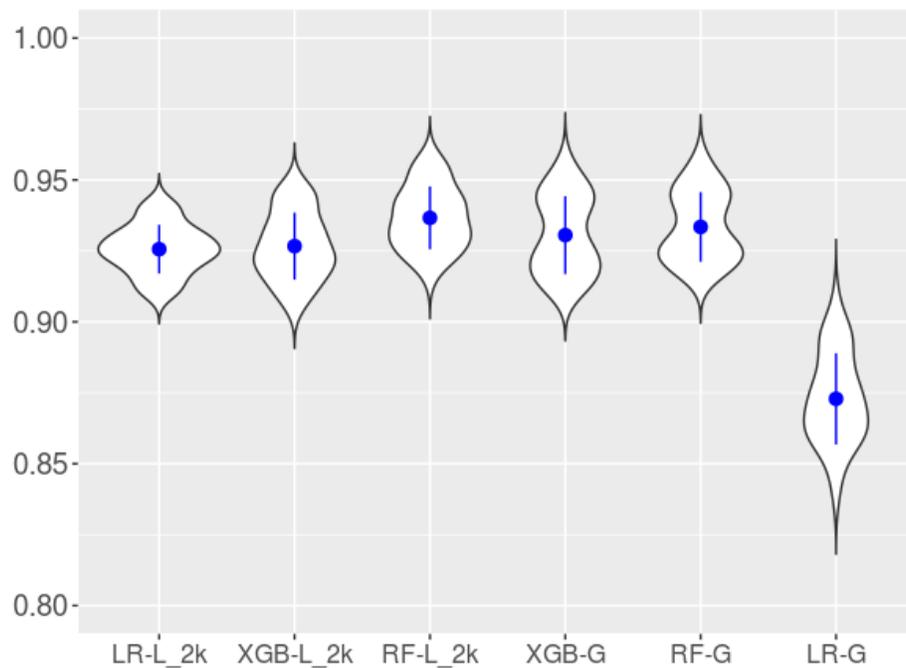
Table 5 summarizes average performance and Figure 20 and Figure 21 depicts.

RF-L_2k is the best performing classifier in terms of average performance.

Table 5: Performance of Local vs Global Classifiers

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, , bold indicate the best classifier for the specific snapshot)

| Model | Mean AUC | Standard Deviation AUC | Mean H-Measure | Standard Deviation H- Measure |
|--------------|---------------------|-----------------------------------|---------------------------|--|
| LR-L_2k | 0.9256 | 0.0086 | 0.6360 | 0.0278 |
| XGB-L_2k | 0.9267 | 0.0118 | 0.6445 | 0.0368 |
| RF-L_2k | 0.9366 | 0.0111 | 0.6695 | 0.0351 |
| LR-G | 0.8729 | 0.0161 | 0.4987 | 0.0344 |
| XGB-G | 0.9306 | 0.0138 | 0.6435 | 0.0382 |
| RF-G | 0.9334 | 0.0123 | 0.6588 | 0.0348 |

*Figure 20: AUC of Local vs Global Classifiers (y-axis not starting from zero)*

LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN)

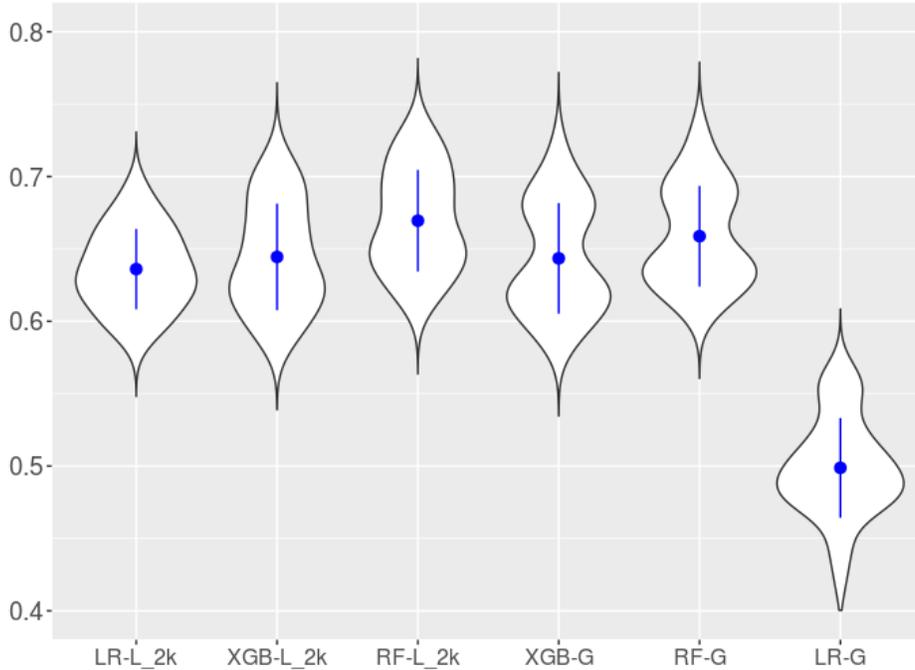


Figure 21: AUC of Local vs Global Classifiers (y-axis not starting from zero)

LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN

Displaying the average ranking of the local and global models used (Figure 22), we observe two things:

- The overall order is grouped by the algorithm used: 1-RF, 2-XGB, 3-LR
- Within each group local model outperforms its corresponding global

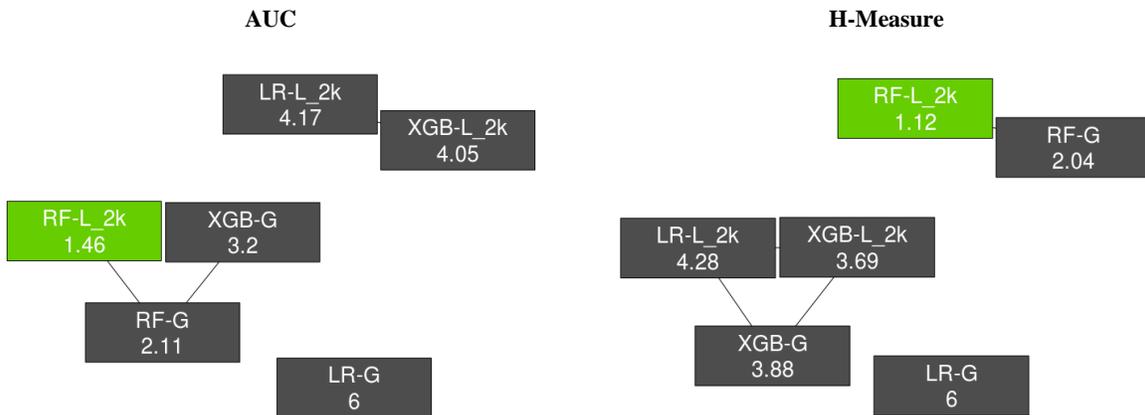


Figure 22: Graph of rankings for global models

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain.)

Analyzing the statistical differences (Figure 23 and Figure 24) 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: 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 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.

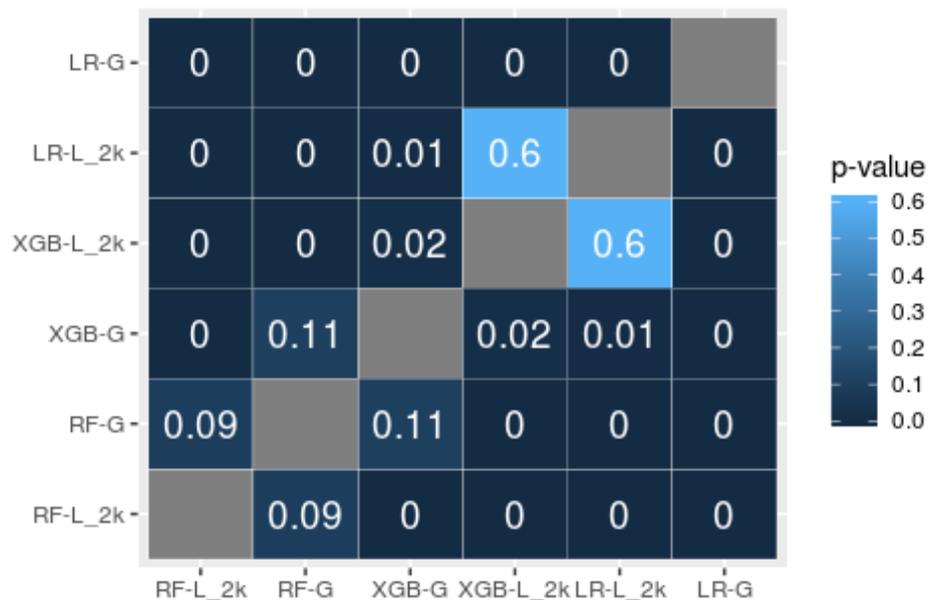


Figure 23: AUC based statistical differences of Local vs Global Classifiers (p-value matrix)

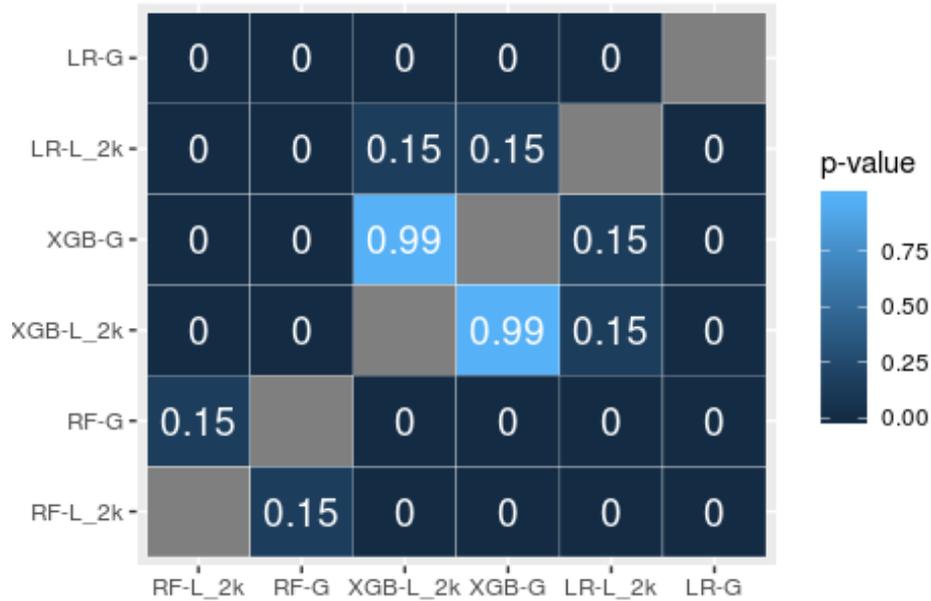


Figure 24: **H-Measure** based statistical differences of Local vs Global Classifiers (p-value matrix)

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 (Figure 25) 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.

To test for statistical differences between all classifiers (i.e. the case of multiple methods on multiple data sets as noted in Demsar (2006)) we use Friedman’s Aligned Rank Test (García et al., 2010) to assess all the pairwise differences between algorithms and then correct the p-values for multiple testing (Figure 6 visualizes the results in matrix format).

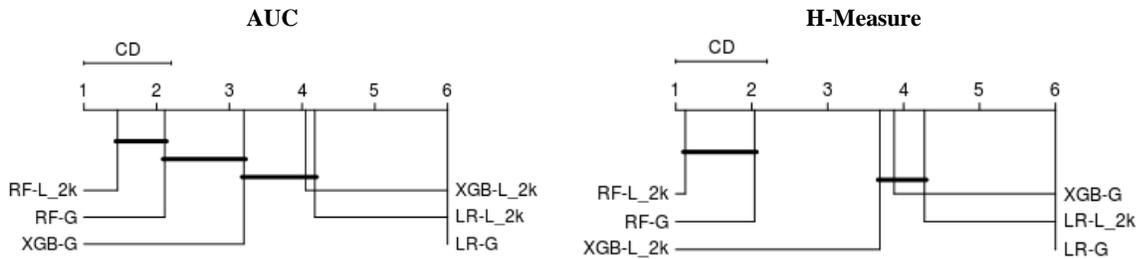


Figure 25: Critical Distances between local and global classifiers

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN)

Thus, it is 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.

4.3 Random regions of competence vs kNNs

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 \mathbf{x} its local region N_x is a set of randomly selected training cases, instead of employing the kNNs scheme. Appendix D: Detailed Results *Table A-15* provided in *Table A-15* whereas the following Figure 26 and Figure 27 highlight the fact that selecting local regions through kNNs does makes a difference and the performance gain

with respect to a random choice of regions is statistical significant. 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 chosen by their similarity (in terms of a distance metric) to the query instance.

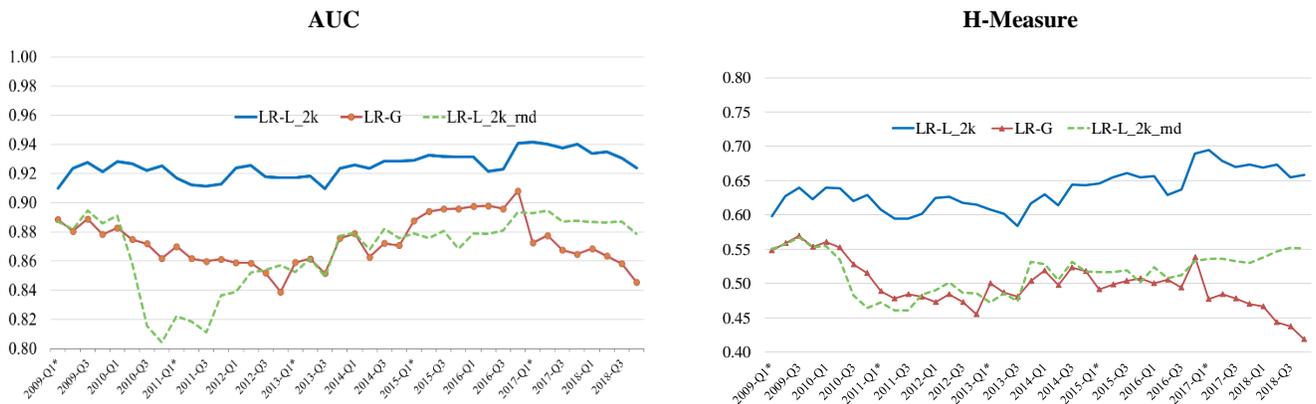


Figure 26: kNNs vs random regions (different y-axis scales)
 (LR=Logistic Regression, G=Global Classifier, 2k=2000 for kNN, *= training snapshot for global LR)

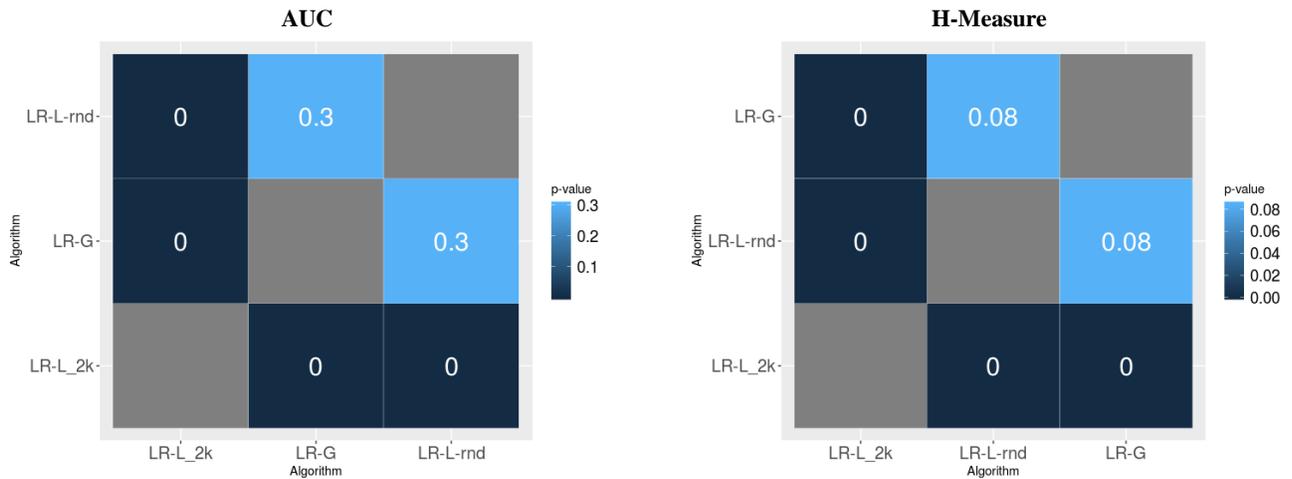


Figure 27: Statistical differences of kNNs vs random regions

5 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, we 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% (AUC) observed in various credit scoring applications of ML/AI found in literature (Addo et al., 2018; Albanesi & Vamossy, 2019; Alonso & Carbó, 2020; Gunnarsson et al., 2021; Hamori et al., 2018; Kvamme et al., 2018). However, it is quite important to observe that the performance of Local LR is on par with RF and XGB. Thus, the performance of Local LR (LR_L) does not differ statistically from the ML algorithms, contrasting the case of global LR (LR_G) which is vastly outranked and outperformed. The gain, when comparing the AUC of these classifiers to the “baseline” global LR, is within the range of 6%-8% (Table 6), which is well within

the empirical range observed in other studies (Alonso & Carbó, 2020) comparing ML algorithms to the basic logistic regression in credit scoring.

Table 6: Performance of Local vs Global Classifiers

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=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% |

Another important observation is that the choice of feature selection method (using in ML models the same variables which were used for LR based on their Information Value) affects negatively ML performance. This is quite understandable as ML methods build explicitly on exploring the entire feature space and therefore constricting their feature space does not allow them to capture all the inherent dynamics.

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. Also we observed that the distance used plays an important role: Although both LR_L methods based on different distance metric outperform LR_G, Euclidean distance is in all cases better than Mahalanobis. This is quite interesting and it may be a result of the way that Mahalanobis

distance works based on the covariance matrix. However, this is a good point for further research.

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 (Guégan & Hassani, 2018; Kiritz & Sarfati, 2018; Morini, 2011) 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 (Chawla et al., 2002), G-SMOTE or G-SOMO (Camacho et al., 2022; Douzas et al., 2021) or RUSBoost (Seiffert et al., 2010) for local sampling considering the highly imbalancing nature of credit datasets (Bischi et al., 2016; He et al., 2018) 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 (Wang et al., 2015; Wang et al., 2017);
- usage of additional distance metrics (e.g., Manhattan)
- usage of different algorithms for choosing local regions instead of the basic kNNs, such as Reduced Minority kNNs (Melo Junior et al., 2020).

Appendix A: Alternative Data

Table A-7: Alternative data in Credit Scoring Source: (ICCR, 2019)

| Data category | Data type | Credit scoring application |
|---------------|-------------------------|--|
| Traditional | Bank transactional data | Records of late payments on current and past credit, current loan amounts and loan purpose, credit history |
| Traditional | Credit bureau checks | Number of credit inquiries |
| Traditional | Commercial data | Financial statements, number of working capital loans, and others |
| Alternative | Utilities data | Steady records of on-time payments as possible consideration as an indicator of creditworthiness |
| Alternative | Social media | Social media data with possible insights on consumer's lifestyle |
| Alternative | Mobile applications | Mobile payment systems with possible view on the consumer's behavior |
| Alternative | Online transactions | Granular transactional data with possible detailed insights on spending patterns |
| Alternative | Behavioral data | Psychometrics, form filling |

Table A-8: Types of data used in Credit Scoring Source: (ICCR, 2019)

| |
|--|
| <p>Examples of factors are as follows:</p> <ul style="list-style-type: none"> • Payment history: A record of late payments on current and past credit accounts may have an adverse effect on an individual's score. Payments on time and in full may improve the score. • Public records: Matters of public record such as bankruptcies, judgments, and collection items may impact the score. • Amount owed and loan purpose: High levels of debt may impact the score. The purpose of the loan and the type of CSP may also be linked to creditworthiness. • Length of credit history and length of time at address: Length of credit history and time at current address are associated with creditworthiness. • New accounts: Opening multiple new credit accounts in a short period of time may impact the score. • Credit bureau checks: Whenever a request for a credit report is made, the inquiry is recorded. Recent inquiries may impact the score. • Social media data: Social media data may provide insights into a consumer's lifestyle, indicating credit worthiness. • Mobile data: Mobile data may provide granular information and insights into consumer behavior. • Utilities data: A steady record of payments may contribute to an individual's credit score. • Commercial data: Financial statements, operational information, and working capital loans may indicate the creditworthiness of businesses. • Macroeconomic data: A change in the macroeconomy (that is, a change in the unemployment rate or GDP of a region) may impact the credit scores of consumers and businesses in that region. |
|--|

Appendix B: List of Variables

| | Feature | Description |
|----|--------------------|--|
| 1 | RH_N_NUM | Number of HOUSING loans |
| 2 | RR_N_NUM | Number of RETAIL |
| 3 | RS_N_NUM | Number of RESTRUCTURING (ρυθμίσεις) |
| 4 | R_N_NUMGR | Number of loans as guarantor |
| 5 | RH_N_DEL_1_MAX | Max delinquency (months in arrears) of HOUSING |
| 6 | RH_N_DEL_3_MAX | LOANS , for the last 1 month (current delinquency), |
| 7 | RH_N_DEL_6_MAX | 3,6,12, 18 months |
| 8 | RH_N_DEL_12_MAX | |
| 9 | RH_N_DEL_18_MAX | |
| 10 | RR_N_DEL_1_MAX | Max delinquency (months in arrears) of RETAIL |
| 11 | RR_N_DEL_3_MAX | LOANS for the last 1 month (current delinquency), |
| 12 | RR_N_DEL_6_MAX | 3,6,12, 18 months |
| 13 | RR_N_DEL_12_MAX | |
| 14 | RR_N_DEL_18_MAX | |
| 15 | RS_N_DEL_1_MAX | Max delinquency (months in arrears) for |
| 16 | RS_N_DEL_3_MAX | RESTRUCTURING LOANS for the last 1 month |
| 17 | RS_N_DEL_6_MAX | (current delinquency), 3,6,12, 18 months |
| 18 | RS_N_DEL_12_MAX | |
| 19 | RS_N_DEL_18_MAX | |
| 20 | R_N_DELGR_MAX | Max. current delinquency (months in arrears) for all loans as GUARANTOR |
| 21 | RH_A_CURBAL_1_MAX | MAX of CURRENT BALANCE of HOUSING |
| 22 | RH_A_CURBAL_3_MAX | LOANS for the last 1 month (current month), |
| 23 | RH_A_CURBAL_6_MAX | 3,6,12, 18 months |
| 24 | RH_A_CURBAL_12_MAX | |
| 25 | RH_A_CURBAL_18_MAX | |
| 26 | RR_A_CURBAL_1_MAX | MAX of CURRENT BALANCE of RETAIL |
| 27 | RR_A_CURBAL_3_MAX | LOANS for the last 1 month (current month), |
| 28 | RR_A_CURBAL_6_MAX | 3,6,12, 18 months |
| 29 | RR_A_CURBAL_12_MAX | |
| 30 | RR_A_CURBAL_18_MAX | |
| 31 | RS_A_CURBAL_1_MAX | MAX of CURRENT BALANCE of |
| 32 | RS_A_CURBAL_3_MAX | RESTRUCTURING LOANS for the last 1 month |
| 33 | RS_A_CURBAL_6_MAX | (current month), 3,6,12, 18 months |
| 34 | RS_A_CURBAL_12_MAX | |
| 35 | RS_A_CURBAL_18_MAX | |
| 36 | RH_A_DELBAL_1_MAX | MAX of Delinquent BALANCE of HOUSING |
| 37 | RH_A_DELBAL_3_MAX | LOANS for the last 1 month (current month), |
| 38 | RH_A_DELBAL_6_MAX | 3,6,12, 18 months |
| 39 | RH_A_DELBAL_12_MAX | |
| 40 | RH_A_DELBAL_18_MAX | |
| 41 | RR_A_DELBAL_1_MAX | MAX of Delinquent BALANCE of RETAIL |
| 42 | RR_A_DELBAL_3_MAX | LOANS for the last 1 month (current month), |
| 43 | RR_A_DELBAL_6_MAX | 3,6,12, 18 months |
| 44 | RR_A_DELBAL_12_MAX | |

| | Feature | Description |
|----|-------------------------|---|
| 45 | RR_A_DELBAL_18_MAX | |
| 46 | RS_A_DELBAL_1_MAX | MAX of Delinquent BALANCE of |
| 47 | RS_A_DELBAL_3_MAX | RESTRUCTURING LOANS for the last 1 month |
| 48 | RS_A_DELBAL_6_MAX | (current month), 3,6,12, 18 months |
| 49 | RS_A_DELBAL_12_MAX | |
| 50 | RS_A_DELBAL_18_MAX | |
| 51 | RH_A_CLIM_1_SUM | Current Credit Limit (for revolving = credit limit, for rest = approval amount), Housing loans |
| 52 | RR_A_CLIM_1_SUM | Current Credit Limit (for revolving = credit limit, for rest = approval amount), retail loans |
| 53 | RS_A_CLIM_1_SUM | Current Credit Limit (for revolving = credit limit, for rest = approval amount), restructuring loans |
| 54 | R_R_BALRATIO_1_MAX | Max. Ratio of [Current delinquent balance / Current |
| 55 | R_R_BALRATIO_3_MAX | balance] among all loans , for the last 1 month |
| 56 | R_R_BALRATIO_6_MAX | (current month), 3,6,12, 18 months |
| 57 | R_R_BALRATIO_12_MAX | |
| 58 | R_R_BALRATIO_18_MAX | |
| 59 | RH_R_BALRATIO_1_MAX | Max. Ratio of [Current delinquent balance / Current |
| 60 | RH_R_BALRATIO_3_MAX | balance] // Housing Loans, for the last 1 month |
| 61 | RH_R_BALRATIO_6_MAX | (current month), 3,6,12, 18 months |
| 62 | RH_R_BALRATIO_12_MAX | |
| 63 | RH_R_BALRATIO_18_MAX | |
| 64 | RR_R_BALRATIO_1_MAX | Max. Ratio of [Current delinquent balance / Current |
| 65 | RR_R_BALRATIO_3_MAX | balance] // Retail Loans, for the last 1 month |
| 66 | RR_R_BALRATIO_6_MAX | (current month), 3,6,12, 18 months |
| 67 | RR_R_BALRATIO_12_MAX | |
| 68 | RR_R_BALRATIO_18_MAX | |
| 69 | RS_R_BALRATIO_1_MAX | Max. Ratio of [Current delinquent balance / Current |
| 70 | RS_R_BALRATIO_3_MAX | balance] // Restructuring Loans, for the last 1 |
| 71 | RS_R_BALRATIO_6_MAX | month (current month), 3,6,12, 18 months |
| 72 | RS_R_BALRATIO_12_MAX | |
| 73 | RS_R_BALRATIO_18_MAX | |
| 74 | R_R_UTILIZATION_1_MAX | Max. Utilization =[Current balance / Credit limit] |
| 75 | R_R_UTILIZATION_3_MAX | // all loans, for the last 1 month (current month), |
| 76 | R_R_UTILIZATION_6_MAX | 3,6,12, 18 months for the last 1 month (current |
| 77 | R_R_UTILIZATION_12_MAX | month), 3,6,12, 18 months |
| 78 | R_R_UTILIZATION_18_MAX | |
| 79 | RH_R_UTILIZATION_1_MAX | Max. Utilization =[Current balance / Credit limit] |
| 80 | RH_R_UTILIZATION_3_MAX | // housing loans, for the last 1 month (current |
| 81 | RH_R_UTILIZATION_6_MAX | month), 3,6,12, 18 months for the last 1 month |
| 82 | RH_R_UTILIZATION_12_MAX | (current month), 3,6,12, 18 months |
| 83 | RH_R_UTILIZATION_18_MAX | |
| 84 | RR_R_UTILIZATION_1_MAX | Max. Utilization =[Current balance / Credit limit] |
| 85 | RR_R_UTILIZATION_3_MAX | // retail loans, for the last 1 month (current month), |
| 86 | RR_R_UTILIZATION_6_MAX | 3,6,12, 18 months for the last 1 month (current |
| 87 | RR_R_UTILIZATION_12_MAX | month), 3,6,12, 18 months |

| | Feature | Description |
|-----|----------------------------|--|
| 88 | RR_R_UTILIZATION_18_MAX | |
| 89 | RS_R_UTILIZATION_1_MAX | Max. Utilization =[Current balance / Credit limit] |
| 90 | RS_R_UTILIZATION_3_MAX | // restructuring loans, for the last 1 month (current |
| 91 | RS_R_UTILIZATION_6_MAX | month), 3,6,12, 18 months for the last 1 month |
| 92 | RS_R_UTILIZATION_12_MAX | (current month), 3,6,12, 18 months |
| 93 | RS_R_UTILIZATION_18_MAX | |
| 94 | R_R_CONSEC12_UTIL100_MAX | Maximum Number of Consecutive Months with over 100% of Percent Credit Utilization in last 12 Months |
| 95 | R_R_CONSEC6_UTIL100_MAX | Maximum Number of Consecutive Months with over 100% of Percent Credit Utilization in last 6 Months |
| 96 | R_R_NUM6_UTIL90_MAX | Total Number of Months with over 90% of Percent Credit Utilization in last 6 months |
| 97 | RR_R_CONSEC12_UTILINCR_MAX | Number of Months with Consecutive Increase of Maximum Percent Credit Utilization in last 12 Months // retail loans |
| 98 | RR_R_CONSEC6_UTILINCR_MAX | Number of Months with Consecutive Increase of Maximum Percent Credit Utilization in last 6 Months // retail loans |
| 99 | R_T_AGEDIFF | Χρόνος μεταξύ νεότερης & παλιότερης χορήγησης |
| 100 | RH_T_AGENEW | Νεότερη χορήγηση -HOUSING |
| 101 | RR_T_AGENEW | Νεότερη χορήγηση -Retail |
| 102 | RS_T_AGENEW | Νεότερη χορήγηση -Restructured |
| 103 | RH_T_18MOS1P | Months Since 1+ months delinquency in last 18 months - Housing |
| 104 | RR_T_18MOS1P | Months Since 1+ months delinquency in last 18 months - Retail |
| 105 | RS_T_18MOS1P | Months Since 1+ months delinquency in last 18 months - Restructured |
| 106 | RH_T_18MOS2P | Months Since 2+ months delinquency in last 18 months - Housing |
| 107 | RR_T_18MOS2P | Months Since 2+ months delinquency in last 18 months - Retail |
| 108 | RS_T_18MOS2P | Months Since 2+ months delinquency in last 18 months - Restructured |
| 109 | RH_N_6OCC1P | Number of Occurrences of Delinquency 1+months - Last 6,12,18 Months //HOUSING |
| 110 | RH_N_12OCC1P | |
| 111 | RH_N_18OCC1P | |
| 112 | RR_N_6 OCC1P | Number of Occurrences of Delinquency 1+months - Last 6,12,18 Months //Retail |
| 113 | RR_N_6 OCC1P | |
| 114 | RR_N_18 OCC1P | |
| 115 | RS_N_6OCC1P | Number of Occurrences of Delinquency 1+months - Last 6,12,18 Months// Restructured |
| 116 | RS_N_6OCC1P | |
| 117 | RS_N_6OCC1P | |
| 118 | D_T_NEWSET | Newest (in months) settled NEGATIVE excluding mortgages |

| | Feature | Description |
|-----|----------------|--|
| 119 | D_T_NEWUNSET | Newest (in months) unsettled NEGATIVE excluding mortgages |
| 120 | D_N_MORTGAGE | Number of mortgages |
| 121 | D_N_NEGATIVE | Number of negative excluding mortgages |
| 122 | I_N_1ALL | No. of inquiries last 1 month |
| 123 | I_N_3ALL | No. of inquiries last 3 months |
| 124 | I_N_12ALL | No. of inquiries last 12 months |
| 125 | I_T_OLDEST | Months since oldest inquiry |

Skim summary statistics
n obs: 3200000 (2009q1 - 2018q12)
n variables: 125

— Variable type:integer

| variable | missing | complete | n | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|--------------------------|---------|----------|---------|--------|-------|----|-----|-----|-----|------|------|
| ALLRCS | 0 | 3200000 | 3200000 | 3.78 | 3.16 | 1 | 2 | 3 | 5 | 131 | |
| D_N_MORTGAGE | 2771606 | 428394 | 3200000 | 1.34 | 0.78 | 0 | 1 | 1 | 2 | 44 | |
| D_N_NEGATIVE | 2771606 | 428394 | 3200000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D_T_NEWSET | 3189510 | 10490 | 3200000 | 59.72 | 32.34 | 12 | 30 | 59 | 85 | 220 | |
| D_T_NEWUNSET | 3200000 | 0 | 3200000 | NaN | NA | NA | NA | NA | NA | NA | |
| I_N_11ALL | 2023992 | 1176008 | 3200000 | 3.21 | 4.05 | 1 | 1 | 2 | 4 | 497 | |
| I_N_1ALL | 2023992 | 1176008 | 3200000 | 0.27 | 0.77 | 0 | 0 | 0 | 0 | 56 | |
| I_N_3ALL | 2023992 | 1176008 | 3200000 | 0.77 | 1.51 | 0 | 0 | 0 | 1 | 149 | |
| I_T_OLDEST | 2023992 | 1176008 | 3200000 | 7.99 | 3.36 | 1 | 5 | 9 | 11 | 12 | |
| R_N_DELGR_MAX | 2701541 | 498459 | 3200000 | 0.33 | 2.04 | -1 | -1 | 0 | 0 | 9 | |
| R_N_NUMGR | 0 | 3200000 | 3200000 | 0.24 | 0.7 | 0 | 0 | 0 | 0 | 34 | |
| R_R_CONSEC12_UTIL100_MAX | 0 | 3200000 | 3200000 | 0.57 | 2.01 | 0 | 0 | 0 | 0 | 12 | |
| R_R_CONSEC6_UTIL100_MAX | 0 | 3200000 | 3200000 | 0.32 | 1.15 | 0 | 0 | 0 | 0 | 6 | |
| R_R_NUM6_UTIL90_MAX | 0 | 3200000 | 3200000 | 1.2 | 2.23 | 0 | 0 | 0 | 1 | 6 | |
| R_T_AGEDIFF | 0 | 3200000 | 3200000 | 73.1 | 96.68 | 0 | 0 | 48 | 112 | 1000 | |
| RH_N_12OCC1P | 2212980 | 987020 | 3200000 | 0.9 | 2.34 | 0 | 0 | 0 | 0 | 12 | |
| RH_N_18OCC1P | 2212980 | 987020 | 3200000 | 1.3 | 3.32 | 0 | 0 | 0 | 0 | 18 | |
| RH_N_6OCC1P | 2212980 | 987020 | 3200000 | 0.46 | 1.27 | 0 | 0 | 0 | 0 | 6 | |
| RH_N_DEL_1_MAX | 2212980 | 987020 | 3200000 | 0.01 | 0.7 | -1 | 0 | 0 | 0 | 9 | |
| RH_N_DEL_12_MAX | 2212980 | 987020 | 3200000 | 0.39 | 1.39 | -1 | 0 | 0 | 0 | 9 | |
| RH_N_DEL_18_MAX | 2212980 | 987020 | 3200000 | 0.5 | 1.53 | -1 | 0 | 0 | 0 | 9 | |
| RH_N_DEL_3_MAX | 2212980 | 987020 | 3200000 | 0.12 | 0.96 | -1 | 0 | 0 | 0 | 9 | |
| RH_N_DEL_6_MAX | 2212980 | 987020 | 3200000 | 0.23 | 1.17 | -1 | 0 | 0 | 0 | 9 | |
| RH_N_NUM | 0 | 3200000 | 3200000 | 0.45 | 0.82 | 0 | 0 | 0 | 1 | 29 | |
| RH_T_18MOS1P | 2957066 | 242934 | 3200000 | 5.48 | 5.04 | 1 | 1 | 3 | 9 | 18 | |
| RH_T_18MOS2P | 3086648 | 113352 | 3200000 | 6.05 | 5.19 | 1 | 1 | 4 | 10 | 18 | |
| RH_T_AGENEW | 2212980 | 987020 | 3200000 | 102.36 | 58.86 | -1 | 57 | 98 | 139 | 999 | |
| RR_N_12OCC1P | 268960 | 2931040 | 3200000 | 1 | 2.51 | 0 | 0 | 0 | 0 | 12 | |
| RR_N_18OCC1P | 268960 | 2931040 | 3200000 | 1.46 | 3.55 | 0 | 0 | 0 | 1 | 18 | |
| RR_N_6OCC1P | 268960 | 2931040 | 3200000 | 0.52 | 1.36 | 0 | 0 | 0 | 0 | 6 | |
| RR_N_DEL_1_MAX | 268960 | 2931040 | 3200000 | 0.16 | 1.05 | -1 | 0 | 0 | 0 | 9 | |
| RR_N_DEL_12_MAX | 268960 | 2931040 | 3200000 | 0.52 | 1.56 | -1 | 0 | 0 | 0 | 9 | |
| RR_N_DEL_18_MAX | 268960 | 2931040 | 3200000 | 0.62 | 1.68 | -1 | 0 | 0 | 1 | 9 | |
| RR_N_DEL_3_MAX | 268960 | 2931040 | 3200000 | 0.27 | 1.21 | -1 | 0 | 0 | 0 | 9 | |
| RR_N_DEL_6_MAX | 268960 | 2931040 | 3200000 | 0.37 | 1.36 | -1 | 0 | 0 | 0 | 9 | |

| | | | | | | | | | | | |
|----------------------------|---------|---------|---------|-------|-------|----|----|----|----|-----|--|
| RR_N_NUM | 0 | 3200000 | 3200000 | 2.99 | 2.78 | 0 | 1 | 2 | 4 | 131 | |
| RR_R_CONSEC12_UTILINCR_MAX | 268960 | 2931040 | 3200000 | 2.05 | 1.97 | 0 | 1 | 2 | 3 | 12 | |
| RR_R_CONSEC6_UTILINCR_MAX | 268960 | 2931040 | 3200000 | 1.58 | 1.37 | 0 | 1 | 1 | 2 | 6 | |
| RR_T_18MOS1P | 2436257 | 763743 | 3200000 | 5.37 | 5.05 | 1 | 1 | 3 | 9 | 18 | |
| RR_T_18MOS2P | 2849625 | 350375 | 3200000 | 5.84 | 5.26 | 1 | 1 | 4 | 10 | 18 | |
| RR_T_AGENEW | 268960 | 2931040 | 3200000 | 68.79 | 66.25 | -1 | 22 | 49 | 95 | 999 | |
| RS_N_12OCC1P | 3025129 | 174871 | 3200000 | 2.92 | 3.83 | 0 | 0 | 1 | 5 | 12 | |
| RS_N_18OCC1P | 3025129 | 174871 | 3200000 | 4.2 | 5.43 | 0 | 0 | 1 | 7 | 18 | |
| RS_N_6OCC1P | 3025129 | 174871 | 3200000 | 1.53 | 2.1 | 0 | 0 | 0 | 3 | 6 | |
| RS_N_DEL_1_MAX | 3025129 | 174871 | 3200000 | 0.57 | 1.62 | -1 | 0 | 0 | 1 | 9 | |
| RS_N_DEL_12_MAX | 3025129 | 174871 | 3200000 | 1.71 | 2.65 | -1 | 0 | 1 | 2 | 9 | |
| RS_N_DEL_18_MAX | 3025129 | 174871 | 3200000 | 2 | 2.82 | -1 | 0 | 1 | 3 | 9 | |
| RS_N_DEL_3_MAX | 3025129 | 174871 | 3200000 | 0.93 | 2 | -1 | 0 | 0 | 1 | 9 | |
| RS_N_DEL_6_MAX | 3025129 | 174871 | 3200000 | 1.28 | 2.33 | -1 | 0 | 0 | 2 | 9 | |
| RS_N_NUM | 0 | 3200000 | 3200000 | 0.097 | 0.53 | 0 | 0 | 0 | 0 | 34 | |
| RS_T_18MOS1P | 3100653 | 99347 | 3200000 | 4.03 | 4.45 | 1 | 1 | 2 | 6 | 18 | |
| RS_T_18MOS2P | 3130307 | 69693 | 3200000 | 4.96 | 4.84 | 1 | 1 | 3 | 8 | 18 | |
| RS_T_AGENEW | 3025129 | 174871 | 3200000 | 38.08 | 35.79 | -1 | 12 | 26 | 55 | 999 | |

— Variable type:numeric

| variable | missing | complete | n | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|------------------------|---------|----------|---------|----------|----------|----|----------|----------|-----------|---------|------|
| R_R_BALRATIO_1_MAX | 358869 | 2841131 | 3200000 | 3 | 14.96 | 0 | 0 | 0 | 461.54 | 999.99 | |
| R_R_BALRATIO_12_MAX | 182489 | 3017511 | 3200000 | 7.3 | 22.4 | 0 | 0 | 0 | 1.35 | 999.99 | |
| R_R_BALRATIO_18_MAX | 170478 | 3029522 | 3200000 | 8.92 | 24.71 | 0 | 0 | 0 | 2.26 | 999.99 | |
| R_R_BALRATIO_3_MAX | 285042 | 2914958 | 3200000 | 4.07 | 16.95 | 0 | 0 | 0 | 461.54 | 999.99 | |
| R_R_BALRATIO_6_MAX | 236354 | 2963646 | 3200000 | 5.29 | 19.16 | 0 | 0 | 0 | 999.99 | 999.99 | |
| R_R_UTILIZATION_1_MAX | 99718 | 3100282 | 3200000 | 51.9 | 67.82 | 0 | 8.41 | 48.01 | 85.66 | 999.99 | |
| R_R_UTILIZATION_12_MAX | 52240 | 3147760 | 3200000 | 64.18 | 71.17 | 0 | 23.88 | 66.88 | 96.56 | 999.99 | |
| R_R_UTILIZATION_18_MAX | 46050 | 3153950 | 3200000 | 68.01 | 72.7 | 0 | 28.82 | 73.03 | 99.31 | 999.99 | |
| R_R_UTILIZATION_3_MAX | 83714 | 3116286 | 3200000 | 55.36 | 68.57 | 0 | 13.05 | 53.18 | 88.59 | 999.99 | |
| R_R_UTILIZATION_6_MAX | 70356 | 3129644 | 3200000 | 58.94 | 69.43 | 0 | 17.5 | 58.79 | 91.83 | 999.99 | |
| RH_A_CLIM_1_SUM | 2337632 | 862368 | 3200000 | 90087.69 | 3e+05 | 0 | 32281.73 | 62900 | 111518.71 | 1e+08 | |
| RH_A_CURBAL_1_MAX | 2337632 | 862368 | 3200000 | 51298.83 | 74720.23 | 0 | 12196.2 | 34069.85 | 67551.93 | 1.2e+07 | |
| RH_A_CURBAL_12_MAX | 2302523 | 897477 | 3200000 | 53587.99 | 77729.79 | 0 | 13495.72 | 36056.84 | 70305.24 | 1.4e+07 | |
| RH_A_CURBAL_18_MAX | 2288498 | 911502 | 3200000 | 54618.96 | 79006.76 | 0 | 14116.5 | 37003.82 | 71578.9 | 1.5e+07 | |
| RH_A_CURBAL_3_MAX | 2328361 | 871639 | 3200000 | 51811.22 | 75301.08 | 0 | 12485.66 | 34518.92 | 68189.08 | 1.3e+07 | |
| RH_A_CURBAL_6_MAX | 2319247 | 880753 | 3200000 | 52460.23 | 76234.93 | 0 | 12853.12 | 35052.78 | 68984.38 | 1.3e+07 | |
| RH_A_DELBAL_1_MAX | 2337632 | 862368 | 3200000 | 209.2 | 6876.51 | 0 | 0 | 0 | 2e+06 | 999.99 | |
| RH_A_DELBAL_12_MAX | 2302523 | 897477 | 3200000 | 451.25 | 9669.31 | 0 | 0 | 0 | 39.66 | 4561912 | |
| RH_A_DELBAL_18_MAX | 2288498 | 911502 | 3200000 | 528.03 | 10168.79 | 0 | 0 | 0 | 110.16 | 4561912 | |
| RH_A_DELBAL_3_MAX | 2328361 | 871639 | 3200000 | 277.83 | 7406.32 | 0 | 0 | 0 | 0 | 2e+06 | |

| | | | | | | | | | | | |
|-------------------------|---------|---------|---------|----------|----------|---|---------|----------|----------|------------|---|
| RH_A_DELBAL_6_MAX | 2319247 | 880753 | 3200000 | 351.19 | 9204.07 | 0 | 0 | 0 | 0 | 4561912 | █ |
| RH_R_BALRATIO_1_MAX | 2340835 | 859165 | 3200000 | 0.75 | 6.94 | 0 | 0 | 0 | 0 | 100 | █ |
| RH_R_BALRATIO_12_MAX | 2306197 | 893803 | 3200000 | 2.24 | 11.5 | 0 | 0 | 0 | 0.12 | 109.12 | █ |
| RH_R_BALRATIO_18_MAX | 2292160 | 907840 | 3200000 | 2.71 | 12.79 | 0 | 0 | 0 | 0.39 | 109.12 | █ |
| RH_R_BALRATIO_3_MAX | 2331698 | 868302 | 3200000 | 1.13 | 8.17 | 0 | 0 | 0 | 0 | 100 | █ |
| RH_R_BALRATIO_6_MAX | 2322826 | 877174 | 3200000 | 1.57 | 9.54 | 0 | 0 | 0 | 0 | 106.81 | █ |
| RH_R_UTILIZATION_1_MAX | 2337634 | 862366 | 3200000 | 67.21 | 77.89 | 0 | 39.53 | 66.92 | 86.58 | 999.99 | █ |
| RH_R_UTILIZATION_12_MAX | 2302528 | 897472 | 3200000 | 71.44 | 79.04 | 0 | 44.45 | 71.47 | 90.12 | 999.99 | █ |
| RH_R_UTILIZATION_18_MAX | 2288503 | 911497 | 3200000 | 73.35 | 79.68 | 0 | 46.86 | 73.55 | 91.66 | 999.99 | █ |
| RH_R_UTILIZATION_3_MAX | 2328363 | 871637 | 3200000 | 68.11 | 78.18 | 0 | 40.61 | 67.89 | 87.33 | 999.99 | █ |
| RH_R_UTILIZATION_6_MAX | 2319252 | 880748 | 3200000 | 69.28 | 78.43 | 0 | 42.01 | 69.18 | 88.33 | 999.99 | █ |
| RR_A_CLIM_1_SUM | 435563 | 2764437 | 3200000 | 10618.79 | 31616.81 | 0 | 2300 | 6000 | 13400 | 4e+07 | █ |
| RR_A_CURBAL_1_MAX | 435563 | 2764437 | 3200000 | 3238.54 | 15090.62 | 0 | 58.96 | 682.68 | 3652.41 | 9592551.33 | █ |
| RR_A_CURBAL_12_MAX | 372458 | 2827542 | 3200000 | 4069.35 | 18538.11 | 0 | 291.45 | 1349.03 | 4964.14 | 1.3e+07 | █ |
| RR_A_CURBAL_18_MAX | 355451 | 2844549 | 3200000 | 4391.62 | 20370.24 | 0 | 366.68 | 1542.21 | 5434.21 | 1.3e+07 | █ |
| RR_A_CURBAL_3_MAX | 418626 | 2781374 | 3200000 | 3444.41 | 15176.43 | 0 | 120.83 | 892.74 | 3964.31 | 9592551.33 | █ |
| RR_A_CURBAL_6_MAX | 401125 | 2798875 | 3200000 | 3678.17 | 15330.81 | 0 | 189.85 | 1074.39 | 4330.47 | 9592551.33 | █ |
| RR_A_DELBAL_1_MAX | 435563 | 2764437 | 3200000 | 93.06 | 1315.71 | 0 | 0 | 0 | 0 | 248569.33 | █ |
| RR_A_DELBAL_12_MAX | 372458 | 2827542 | 3200000 | 171.45 | 8207.13 | 0 | 0 | 0 | 0 | 1.3e+07 | █ |
| RR_A_DELBAL_18_MAX | 355451 | 2844549 | 3200000 | 201.74 | 11461.37 | 0 | 0 | 0 | 29.35 | 1.3e+07 | █ |
| RR_A_DELBAL_3_MAX | 418626 | 2781374 | 3200000 | 112.72 | 1441.27 | 0 | 0 | 0 | 0 | 593303.5 | █ |
| RR_A_DELBAL_6_MAX | 401125 | 2798875 | 3200000 | 134.03 | 1551.66 | 0 | 0 | 0 | 0 | 593303.5 | █ |
| RR_R_BALRATIO_1_MAX | 882935 | 2317065 | 3200000 | 3.23 | 15.59 | 0 | 0 | 0 | 0 | 461.54 | █ |
| RR_R_BALRATIO_12_MAX | 648329 | 2551671 | 3200000 | 7.64 | 23.01 | 0 | 0 | 0 | 1.4 | 999.99 | █ |
| RR_R_BALRATIO_18_MAX | 615449 | 2584551 | 3200000 | 9.31 | 25.32 | 0 | 0 | 0 | 2.46 | 999.99 | █ |
| RR_R_BALRATIO_3_MAX | 793318 | 2406682 | 3200000 | 4.33 | 17.57 | 0 | 0 | 0 | 0 | 461.54 | █ |
| RR_R_BALRATIO_6_MAX | 727801 | 2472199 | 3200000 | 5.58 | 19.77 | 0 | 0 | 0 | 0 | 999.99 | █ |
| RR_R_UTILIZATION_1_MAX | 524569 | 2675431 | 3200000 | 40.4 | 50.61 | 0 | 2.06 | 26.06 | 75.74 | 999.99 | █ |
| RR_R_UTILIZATION_12_MAX | 437360 | 2762640 | 3200000 | 54.45 | 56.45 | 0 | 11.34 | 51.41 | 94.22 | 999.99 | █ |
| RR_R_UTILIZATION_18_MAX | 412607 | 2787393 | 3200000 | 58.59 | 58.58 | 0 | 14.59 | 59.37 | 98.32 | 999.99 | █ |
| RR_R_UTILIZATION_3_MAX | 501284 | 2698716 | 3200000 | 44.48 | 51.89 | 0 | 4.59 | 33.71 | 81.01 | 999.99 | █ |
| RR_R_UTILIZATION_6_MAX | 476305 | 2723695 | 3200000 | 48.6 | 53.54 | 0 | 7.28 | 41.01 | 86.5 | 999.99 | █ |
| RS_A_CLIM_1_SUM | 3039443 | 160557 | 3200000 | 46802.67 | 1e+05 | 0 | 7583.89 | 18600 | 50000 | 1.3e+07 | █ |
| RS_A_CURBAL_1_MAX | 3039443 | 160557 | 3200000 | 31115.44 | 55603.63 | 0 | 5854.73 | 14776.11 | 35923.2 | 3342022.89 | █ |
| RS_A_CURBAL_12_MAX | 3033233 | 166767 | 3200000 | 32157.8 | 57484.08 | 0 | 6298.91 | 15482.15 | 36983.72 | 3478702.4 | █ |
| RS_A_CURBAL_18_MAX | 3031883 | 168117 | 3200000 | 32491.65 | 58018.09 | 0 | 6466.38 | 15700 | 37320.57 | 3540281.09 | █ |
| RS_A_CURBAL_3_MAX | 3037078 | 162922 | 3200000 | 31497.29 | 56611.6 | 0 | 5976.16 | 15002.67 | 36321.24 | 3343346.93 | █ |
| RS_A_CURBAL_6_MAX | 3035327 | 164673 | 3200000 | 31718.71 | 56902.13 | 0 | 6087.52 | 15157.2 | 36571.83 | 3343346.93 | █ |
| RS_A_DELBAL_1_MAX | 3039443 | 160557 | 3200000 | 580.94 | 5943.98 | 0 | 0 | 0 | 76.57 | 721648.14 | █ |
| RS_A_DELBAL_12_MAX | 3033233 | 166767 | 3200000 | 1334.6 | 9664.2 | 0 | 0 | 72.91 | 345.39 | 1590429.57 | █ |
| RS_A_DELBAL_18_MAX | 3031883 | 168117 | 3200000 | 1561.19 | 10370.71 | 0 | 0 | 106.07 | 449.88 | 1590429.57 | █ |

| | | | | | | | | | | | |
|-------------------------|---------|--------|---------|---------|---------|---|-------|-------|--------|------------|---|
| RS_A_DELBAL_3_MAX | 3037078 | 162922 | 3200000 | 768.68 | 6680.09 | 0 | 0 | 0 | 150.33 | 721648.14 | █ |
| RS_A_DELBAL_6_MAX | 3035327 | 164673 | 3200000 | 1005.88 | 8716.38 | 0 | 0 | 22.51 | 224.35 | 1590429.57 | █ |
| RS_R_BALRATIO_1_MAX | 3039860 | 160140 | 3200000 | 3.65 | 16.6 | 0 | 0 | 0 | 0.6 | 100 | █ |
| RS_R_BALRATIO_12_MAX | 3033539 | 166461 | 3200000 | 7.9 | 22.9 | 0 | 0 | 0.55 | 2.93 | 126.58 | █ |
| RS_R_BALRATIO_18_MAX | 3032130 | 167870 | 3200000 | 9.26 | 24.63 | 0 | 0 | 0.85 | 3.7 | 128.7 | █ |
| RS_R_BALRATIO_3_MAX | 3037380 | 162620 | 3200000 | 4.77 | 18.33 | 0 | 0 | 0 | 1.36 | 124.15 | █ |
| RS_R_BALRATIO_6_MAX | 3035646 | 164354 | 3200000 | 6.04 | 20.27 | 0 | 0 | 0.07 | 1.99 | 124.83 | █ |
| RS_R_UTILIZATION_1_MAX | 3039454 | 160546 | 3200000 | 103.46 | 118.86 | 0 | 81.22 | 96.29 | 100 | 999.99 | █ |
| RS_R_UTILIZATION_12_MAX | 3033245 | 166755 | 3200000 | 108.72 | 118.46 | 0 | 89.35 | 99.7 | 100.94 | 999.99 | █ |
| RS_R_UTILIZATION_18_MAX | 3031895 | 168105 | 3200000 | 110.48 | 117.75 | 0 | 92.64 | 100 | 101.24 | 999.99 | █ |
| RS_R_UTILIZATION_3_MAX | 3037089 | 162911 | 3200000 | 105.07 | 120.15 | 0 | 82.98 | 97.14 | 100.22 | 999.99 | █ |
| RS_R_UTILIZATION_6_MAX | 3035338 | 164662 | 3200000 | 106.44 | 119.5 | 0 | 85.31 | 98.08 | 100.57 | 999.99 | █ |

Appendix C: Execution Environment

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.

We used default values for the parameters for the calculation of H-measure as defined in the corresponding R-Package.

Below detailed information about the execution environment is provided:

– Session info

```
setting value
version R version 3.5.1 (2018-07-02)
os      Ubuntu 16.04.7 LTS
system  x86_64, linux-gnu
ui      RStudio
language (EN)
collate en_US.UTF-8
tz      Europe/Athens
date    2022-06-28
```

– Packages

| package | * version | date | source |
|--|-----------|------------|----------------|
| <i>'*': whether the package is attached to the search path</i> | | | |
| abind | 1.4-5 | 2016-07-21 | CRAN (R 3.5.1) |
| assertthat | 0.2.0 | 2017-04-11 | CRAN (R 3.5.1) |
| backports | 1.1.2 | 2017-12-13 | CRAN (R 3.5.1) |
| base64url | 1.4 | 2018-05-14 | CRAN (R 3.5.1) |
| BBmisc | 1.11 | 2017-03-10 | CRAN (R 3.5.1) |
| bindr | 0.1.1 | 2018-03-13 | CRAN (R 3.5.1) |
| bindrcpp | 0.2.2 | 2018-03-29 | CRAN (R 3.5.1) |
| BiocGenerics | * 0.26.0 | 2020-03-13 | Bioconductor |
| boot | * 1.3-20 | 2017-08-06 | CRAN (R 3.5.1) |
| Boruta | * 6.0.0 | 2018-07-17 | CRAN (R 3.5.1) |
| broom | 0.5.0 | 2018-07-17 | CRAN (R 3.5.1) |
| caret | * 6.0-80 | 2018-05-26 | CRAN (R 3.5.1) |
| cellranger | 1.1.0 | 2016-07-27 | CRAN (R 3.5.1) |
| checkmate | 1.8.5 | 2017-10-24 | CRAN (R 3.5.1) |
| class | * 7.3-14 | 2015-08-30 | CRAN (R 3.5.1) |
| cli | 1.0.0 | 2017-11-05 | CRAN (R 3.5.1) |
| clisymbols | 1.2.0 | 2017-05-21 | CRAN (R 3.5.1) |
| codetools | 0.2-15 | 2016-10-05 | CRAN (R 3.5.1) |

| | | | | |
|-------------|----------|------------|--------------|-----------|
| colorspace | 1.3-2 | 2016-12-14 | CRAN | (R 3.5.1) |
| crayon | 1.3.4 | 2017-09-16 | CRAN | (R 3.5.1) |
| CVST | 0.2-2 | 2018-05-26 | CRAN | (R 3.5.1) |
| data.table | * 1.11.4 | 2018-05-27 | CRAN | (R 3.5.1) |
| dbscan | * 1.1-2 | 2018-05-19 | CRAN | (R 3.5.1) |
| ddalpha | 1.3.4 | 2018-06-23 | CRAN | (R 3.5.1) |
| DEoptimR | 1.0-8 | 2016-11-19 | CRAN | (R 3.5.1) |
| dgof | * 1.2 | 2013-10-25 | CRAN | (R 3.5.1) |
| digest | 0.6.15 | 2018-01-28 | CRAN | (R 3.5.1) |
| dimRed | 0.1.0 | 2017-05-04 | CRAN | (R 3.5.1) |
| doMC | * 1.3.5 | 2017-12-12 | CRAN | (R 3.5.1) |
| doParallel | 1.0.11 | 2017-09-28 | CRAN | (R 3.5.1) |
| dplyr | * 0.7.6 | 2018-06-29 | CRAN | (R 3.5.1) |
| drake | * 5.3.0 | 2018-07-19 | CRAN | (R 3.5.1) |
| DRR | 0.0.3 | 2018-01-06 | CRAN | (R 3.5.1) |
| e1071 | * 1.7-0 | 2018-07-28 | CRAN | (R 3.5.1) |
| evaluate | 0.11 | 2018-07-17 | CRAN | (R 3.5.1) |
| extrafont | 0.17 | 2014-12-08 | CRAN | (R 3.5.1) |
| extrafontdb | 1.0 | 2012-06-11 | CRAN | (R 3.5.1) |
| fastmatch | 1.1-0 | 2017-01-28 | CRAN | (R 3.5.1) |
| filelock | * 1.0.1 | 2018-02-07 | CRAN | (R 3.5.1) |
| flexclust | * 1.3-5 | 2018-02-14 | CRAN | (R 3.5.1) |
| forcats | * 0.3.0 | 2018-02-19 | CRAN | (R 3.5.1) |
| foreach | * 1.4.4 | 2017-12-12 | CRAN | (R 3.5.1) |
| formatR | 1.5 | 2017-04-25 | CRAN | (R 3.5.1) |
| Formula | 1.2-3 | 2018-05-03 | CRAN | (R 3.5.1) |
| fs | 1.2.5 | 2018-07-30 | CRAN | (R 3.5.1) |
| fst | * 0.8.8 | 2018-06-07 | CRAN | (R 3.5.1) |
| geometry | 0.3-6 | 2015-09-09 | CRAN | (R 3.5.1) |
| ggplot2 | * 3.0.0 | 2018-07-03 | CRAN | (R 3.5.1) |
| glmnet | * 2.0-16 | 2018-04-02 | CRAN | (R 3.5.1) |
| glue | 1.3.0 | 2018-07-17 | CRAN | (R 3.5.1) |
| gower | 0.1.2 | 2017-02-23 | CRAN | (R 3.5.1) |
| graph | * 1.58.2 | 2018-10-09 | Bioconductor | |
| gridExtra | 2.3 | 2017-09-09 | CRAN | (R 3.5.1) |
| gtable | 0.2.0 | 2016-02-26 | CRAN | (R 3.5.1) |
| hashr | * 0.1.0 | 2015-08-06 | CRAN | (R 3.5.1) |
| haven | 1.1.2 | 2018-06-27 | CRAN | (R 3.5.1) |
| hmeasure | * 1.0 | 2012-09-10 | CRAN | (R 3.5.1) |
| hms | 0.4.2 | 2018-03-10 | CRAN | (R 3.5.1) |
| hrbrthemes | * 0.5.0 | 2018-04-24 | CRAN | (R 3.5.1) |
| htmltools | 0.3.6 | 2017-04-28 | CRAN | (R 3.5.1) |
| httr | 1.3.1 | 2017-08-20 | CRAN | (R 3.5.1) |
| igraph | 1.2.2 | 2018-07-27 | CRAN | (R 3.5.1) |
| Information | * 0.0.9 | 2016-04-09 | CRAN | (R 3.5.1) |
| inum | 1.0-0 | 2017-12-12 | CRAN | (R 3.5.1) |
| ipred | 0.9-6 | 2017-03-01 | CRAN | (R 3.5.1) |
| iterators | * 1.0.10 | 2018-08-01 | local | |
| jsonlite | 1.5 | 2017-06-01 | CRAN | (R 3.5.0) |

| | | | |
|--------------|------------|------------|--|
| JuliaCall | * 0.17.1 | 2019-12-08 | Github (Non-Contradiction/JuliaCall@5ed8563) |
| KernelKnn | * 1.0.8 | 2018-01-16 | CRAN (R 3.5.1) |
| kernlab | 0.9-26 | 2018-04-30 | CRAN (R 3.5.1) |
| knitr | 1.20 | 2018-02-20 | CRAN (R 3.5.1) |
| labeling | 0.3 | 2014-08-23 | CRAN (R 3.5.1) |
| lattice | * 0.20-35 | 2017-03-25 | CRAN (R 3.5.1) |
| lava | 1.6.2 | 2018-07-02 | CRAN (R 3.5.1) |
| lazyeval | 0.2.1 | 2017-10-29 | CRAN (R 3.5.1) |
| libcoin | 1.0-1 | 2017-12-13 | CRAN (R 3.5.1) |
| logiBin | * 0.3 | 2018-05-21 | CRAN (R 3.5.1) |
| lubridate | 1.7.4 | 2018-04-11 | CRAN (R 3.5.1) |
| magic | 1.5-8 | 2018-01-26 | CRAN (R 3.5.1) |
| magrittr | * 1.5 | 2014-11-22 | CRAN (R 3.5.1) |
| MASS | * 7.3-50 | 2018-04-30 | CRAN (R 3.5.1) |
| Matrix | * 1.2-14 | 2018-04-13 | CRAN (R 3.5.1) |
| mefa4 | * 0.3-5 | 2018-03-25 | CRAN (R 3.5.1) |
| MLmetrics | * 1.1.1 | 2016-05-13 | CRAN (R 3.5.1) |
| mlr | * 2.12.1 | 2018-03-29 | CRAN (R 3.5.1) |
| ModelMetrics | 1.1.0 | 2016-08-26 | CRAN (R 3.5.1) |
| modelr | 0.1.2 | 2018-05-11 | CRAN (R 3.5.1) |
| modeltools | * 0.2-22 | 2018-07-16 | CRAN (R 3.5.1) |
| munsell | 0.5.0 | 2018-06-12 | CRAN (R 3.5.1) |
| mvtnorm | 1.0-8 | 2018-05-31 | CRAN (R 3.5.1) |
| nlme | 3.1-137 | 2018-04-07 | CRAN (R 3.5.1) |
| nnet | 7.3-12 | 2016-02-02 | CRAN (R 3.5.1) |
| pacman | * 0.4.6 | 2017-05-14 | CRAN (R 3.5.1) |
| parallelMap | 1.3 | 2015-06-10 | CRAN (R 3.5.1) |
| ParamHelpers | * 1.11 | 2018-06-25 | CRAN (R 3.5.1) |
| partykit | 1.2-2 | 2018-06-05 | CRAN (R 3.5.1) |
| pbapply | * 1.3-4 | 2018-01-10 | CRAN (R 3.5.1) |
| pillar | 1.3.0 | 2018-07-14 | CRAN (R 3.5.1) |
| pkgconfig | 2.0.1 | 2017-03-21 | CRAN (R 3.5.1) |
| pls | 2.6-0 | 2016-12-18 | CRAN (R 3.5.1) |
| plyr | 1.8.4 | 2016-06-08 | CRAN (R 3.5.1) |
| pROC | * 1.12.1 | 2018-05-06 | CRAN (R 3.5.1) |
| proclim | 2018.04.18 | 2018-04-18 | CRAN (R 3.5.1) |
| ps | * 1.3.0 | 2018-12-21 | CRAN (R 3.5.1) |
| purrr | * 0.2.5 | 2018-05-29 | CRAN (R 3.5.1) |
| R.methodsS3 | 1.7.1 | 2016-02-16 | CRAN (R 3.5.1) |
| R.oo | 1.22.0 | 2018-04-22 | CRAN (R 3.5.1) |
| R.utils | 2.6.0 | 2017-11-05 | CRAN (R 3.5.1) |
| R6 | 2.2.2 | 2017-06-17 | CRAN (R 3.5.0) |
| ramify | * 0.3.3 | 2016-12-17 | CRAN (R 3.5.1) |
| randomForest | * 4.6-14 | 2018-03-25 | CRAN (R 3.5.1) |
| ranger | * 0.10.1 | 2018-06-04 | CRAN (R 3.5.1) |
| Rcpp | 0.12.18 | 2018-07-23 | CRAN (R 3.5.1) |
| RcppRoll | 0.3.0 | 2018-06-05 | CRAN (R 3.5.1) |
| readr | * 1.1.1 | 2017-05-16 | CRAN (R 3.5.1) |

| | | | |
|---------------|----------|------------|--|
| readxl | * 1.1.0 | 2018-04-20 | CRAN (R 3.5.1) |
| recipes | 0.1.3 | 2018-06-16 | CRAN (R 3.5.1) |
| reshape2 | * 1.4.3 | 2017-12-11 | CRAN (R 3.5.1) |
| RevoUtils | * 11.0.1 | 2018-08-01 | local |
| RevoUtilsMath | * 11.0.0 | 2018-08-01 | local |
| Rgraphviz | * 2.26.0 | 2018-10-30 | Bioconductor |
| rlang | 0.2.1 | 2018-05-30 | CRAN (R 3.5.1) |
| rlist | 0.4.6.1 | 2016-04-04 | CRAN (R 3.5.1) |
| rmarkdown | 1.10 | 2018-06-11 | CRAN (R 3.5.1) |
| robustbase | 0.93-2 | 2018-07-27 | CRAN (R 3.5.1) |
| rpart | 4.1-13 | 2018-02-23 | CRAN (R 3.5.1) |
| rprojroot | 1.3-2 | 2018-01-03 | CRAN (R 3.5.1) |
| rstudioapi | 0.7 | 2017-09-07 | CRAN (R 3.5.1) |
| Rttf2pt1 | 1.3.7 | 2018-06-29 | CRAN (R 3.5.1) |
| rvest | 0.3.2 | 2016-06-17 | CRAN (R 3.5.1) |
| scales | 0.5.0 | 2017-08-24 | CRAN (R 3.5.1) |
| scmamp | * 0.2.55 | 2016-10-21 | CRAN (R 3.5.1) |
| scorecard | * 0.1.8 | 2018-06-12 | CRAN (R 3.5.1) |
| sessioninfo | * 1.0.0 | 2017-06-21 | CRAN (R 3.5.1) |
| sfsmisc | 1.1-2 | 2018-03-05 | CRAN (R 3.5.1) |
| snow | * 0.4-2 | 2016-10-14 | CRAN (R 3.5.1) |
| speedglm | * 0.3-2 | 2017-01-09 | CRAN (R 3.5.1) |
| storr | * 1.2.0 | 2018-05-31 | CRAN (R 3.5.1) |
| stringi | 1.2.4 | 2018-07-20 | CRAN (R 3.5.1) |
| stringr | * 1.3.1 | 2018-05-10 | CRAN (R 3.5.1) |
| strip | * 0.1.1 | 2017-01-13 | CRAN (R 3.5.1) |
| survival | 2.42-3 | 2018-04-16 | CRAN (R 3.5.1) |
| testthat | 2.0.0 | 2017-12-13 | CRAN (R 3.5.1) |
| | | | |
| tibble | * 1.4.2 | 2018-01-22 | CRAN (R 3.5.1) |
| tictoc | * 1.0 | 2014-06-17 | CRAN (R 3.5.1) |
| tidyr | * 0.8.1 | 2018-05-18 | CRAN (R 3.5.1) |
| tidyselect | 0.2.4 | 2018-02-26 | CRAN (R 3.5.1) |
| tidyverse | * 1.2.1 | 2017-11-14 | CRAN (R 3.5.1) |
| timeDate | 3043.102 | 2018-02-21 | CRAN (R 3.5.1) |
| txtq | * 0.0.4 | 2018-06-15 | CRAN (R 3.5.1) |
| viridis | * 0.5.1 | 2018-03-29 | CRAN (R 3.5.1) |
| viridisLite | * 0.3.0 | 2018-02-01 | CRAN (R 3.5.1) |
| withr | 2.1.2 | 2018-03-15 | CRAN (R 3.5.1) |
| xgboost | * 0.71.2 | 2018-06-09 | CRAN (R 3.5.1) |
| xml2 | 1.2.0 | 2018-01-24 | cran (@1.2.0 |
| xxhashlite | * 0.2.1 | 2021-01-05 | Github (coolbutuseless/xxhashlite@32df619) |

Appendix D: Detailed Results

Table A-9: Performance (AUC) for Global Classifiers

LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, SD= Standard deviation, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain)

| | LR_G_n_IV | LR_G_IV | XGB_n_FS | XGB_G_FS | XGB_G_IV | RF_G_n_FS | RF_G_FS | RF_G_IV |
|---------|-----------|---------|----------|----------|----------|-----------|---------|---------|
| 2009-Q1 | 0.8885 | 0.8885 | 0.9158 | 0.9158 | 0.9124 | 0.9193 | 0.9193 | 0.9162 |
| 2009-Q2 | 0.8806 | 0.8806 | 0.9113 | 0.9113 | 0.9167 | 0.9213 | 0.9213 | 0.9198 |
| 2009-Q3 | 0.8889 | 0.8889 | 0.9159 | 0.9159 | 0.9204 | 0.9246 | 0.9246 | 0.9259 |
| 2009-Q4 | 0.8784 | 0.8784 | 0.9170 | 0.9170 | 0.9170 | 0.9214 | 0.9214 | 0.9223 |
| 2010-Q1 | 0.8829 | 0.8829 | 0.9183 | 0.9183 | 0.9193 | 0.9251 | 0.9251 | 0.9270 |
| 2010-Q2 | 0.8749 | 0.8749 | 0.9208 | 0.9208 | 0.9214 | 0.9276 | 0.9276 | 0.9289 |
| 2010-Q3 | 0.8720 | 0.8720 | 0.9118 | 0.9118 | 0.9143 | 0.9198 | 0.9198 | 0.9242 |
| 2010-Q4 | 0.8619 | 0.8619 | 0.9111 | 0.9111 | 0.9143 | 0.9169 | 0.9169 | 0.9215 |
| 2011-Q1 | 0.8498 | 0.8701 | 0.8994 | 0.9246 | 0.9206 | 0.9093 | 0.9265 | 0.9218 |
| 2011-Q2 | 0.8366 | 0.8618 | 0.8962 | 0.9222 | 0.9187 | 0.9028 | 0.9236 | 0.9182 |
| 2011-Q3 | 0.8397 | 0.8599 | 0.8820 | 0.9114 | 0.9113 | 0.8979 | 0.9168 | 0.9125 |
| 2011-Q4 | 0.8393 | 0.8613 | 0.8773 | 0.9166 | 0.9151 | 0.8987 | 0.9196 | 0.9146 |
| 2012-Q1 | 0.8257 | 0.8589 | 0.8875 | 0.9219 | 0.9221 | 0.9096 | 0.9264 | 0.9234 |
| 2012-Q2 | 0.8304 | 0.8587 | 0.8910 | 0.9243 | 0.9232 | 0.9112 | 0.9275 | 0.9234 |
| 2012-Q3 | 0.8233 | 0.8519 | 0.8841 | 0.9209 | 0.9198 | 0.9048 | 0.9251 | 0.9200 |
| 2012-Q4 | 0.8099 | 0.8389 | 0.8814 | 0.9169 | 0.9168 | 0.9032 | 0.9197 | 0.9130 |
| 2013-Q1 | 0.8057 | 0.8591 | 0.8745 | 0.9253 | 0.9209 | 0.9010 | 0.9256 | 0.9182 |
| 2013-Q2 | 0.8138 | 0.8617 | 0.8839 | 0.9237 | 0.9174 | 0.9052 | 0.9248 | 0.9178 |
| 2013-Q3 | 0.8014 | 0.8516 | 0.8687 | 0.9142 | 0.9096 | 0.8915 | 0.9154 | 0.9086 |
| 2013-Q4 | 0.8201 | 0.8757 | 0.8919 | 0.9271 | 0.9224 | 0.9128 | 0.9288 | 0.9239 |
| 2014-Q1 | 0.8227 | 0.8790 | 0.8954 | 0.9299 | 0.9255 | 0.9190 | 0.9318 | 0.9277 |
| 2014-Q2 | 0.8046 | 0.8628 | 0.8875 | 0.9265 | 0.9238 | 0.9115 | 0.9302 | 0.9249 |
| 2014-Q3 | 0.8108 | 0.8722 | 0.8932 | 0.9333 | 0.9332 | 0.9198 | 0.9392 | 0.9343 |
| 2014-Q4 | 0.8012 | 0.8708 | 0.8916 | 0.9330 | 0.9315 | 0.9190 | 0.9369 | 0.9321 |
| 2015-Q1 | 0.8006 | 0.8878 | 0.8911 | 0.9397 | 0.9366 | 0.9167 | 0.9404 | 0.9362 |
| 2015-Q2 | 0.8054 | 0.8941 | 0.9033 | 0.9452 | 0.9417 | 0.9249 | 0.9470 | 0.9432 |
| 2015-Q3 | 0.8123 | 0.8958 | 0.9084 | 0.9457 | 0.9400 | 0.9274 | 0.9455 | 0.9404 |
| 2015-Q4 | 0.8095 | 0.8960 | 0.9046 | 0.9426 | 0.9392 | 0.9288 | 0.9442 | 0.9399 |
| 2016-Q1 | 0.8158 | 0.8975 | 0.9091 | 0.9489 | 0.9450 | 0.9331 | 0.9515 | 0.9481 |
| 2016-Q2 | 0.8113 | 0.8980 | 0.9038 | 0.9428 | 0.9392 | 0.9279 | 0.9454 | 0.9394 |
| 2016-Q3 | 0.8098 | 0.8959 | 0.9004 | 0.9437 | 0.9400 | 0.9251 | 0.9473 | 0.9424 |
| 2016-Q4 | 0.8106 | 0.9082 | 0.9175 | 0.9540 | 0.9509 | 0.9382 | 0.9566 | 0.9526 |
| 2017-Q1 | 0.8109 | 0.8725 | 0.9163 | 0.9559 | 0.9508 | 0.9384 | 0.9571 | 0.9499 |
| 2017-Q2 | 0.8051 | 0.8776 | 0.9124 | 0.9518 | 0.9483 | 0.9355 | 0.9516 | 0.9483 |
| 2017-Q3 | 0.7913 | 0.8676 | 0.8942 | 0.9432 | 0.9384 | 0.9227 | 0.9426 | 0.9368 |
| 2017-Q4 | 0.7787 | 0.8649 | 0.8941 | 0.9439 | 0.9398 | 0.9229 | 0.9451 | 0.9392 |
| 2018-Q1 | 0.7707 | 0.8686 | 0.8994 | 0.9462 | 0.9407 | 0.9278 | 0.9460 | 0.9403 |
| 2018-Q2 | 0.7388 | 0.8635 | 0.9024 | 0.9470 | 0.9427 | 0.9285 | 0.9491 | 0.9454 |
| 2018-Q3 | 0.7707 | 0.8583 | 0.8977 | 0.9402 | 0.9358 | 0.9218 | 0.9389 | 0.9324 |
| 2018-Q4 | 0.7487 | 0.8455 | 0.9009 | 0.9375 | 0.9330 | 0.9228 | 0.9352 | 0.9290 |

Table A-10: Performance (*H-Measure*) for Global Classifiers

LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, G=Global Classifier, SD= Standard deviation, IV=feature selection based on IV, FS=implicit feature selection, n=no retrain)

| | LR_G_n_IV | LR_G_IV | XGB_n_FS | XGB_G_FS | XGB_G_IV | RF_G_n_FS | RF_G_FS | RF_G_IV |
|---------|-----------|---------|----------|----------|----------|-----------|---------|---------|
| 2009-Q1 | 0.8885 | 0.8885 | 0.9158 | 0.9158 | 0.9124 | 0.9193 | 0.9193 | 0.9162 |
| 2009-Q2 | 0.8806 | 0.8806 | 0.9113 | 0.9113 | 0.9167 | 0.9213 | 0.9213 | 0.9198 |
| 2009-Q3 | 0.8889 | 0.8889 | 0.9159 | 0.9159 | 0.9204 | 0.9246 | 0.9246 | 0.9259 |
| 2009-Q4 | 0.8784 | 0.8784 | 0.9170 | 0.9170 | 0.9170 | 0.9214 | 0.9214 | 0.9223 |
| 2010-Q1 | 0.8829 | 0.8829 | 0.9183 | 0.9183 | 0.9193 | 0.9251 | 0.9251 | 0.9270 |
| 2010-Q2 | 0.8749 | 0.8749 | 0.9208 | 0.9208 | 0.9214 | 0.9276 | 0.9276 | 0.9289 |
| 2010-Q3 | 0.8720 | 0.8720 | 0.9118 | 0.9118 | 0.9143 | 0.9198 | 0.9198 | 0.9242 |
| 2010-Q4 | 0.8619 | 0.8619 | 0.9111 | 0.9111 | 0.9143 | 0.9169 | 0.9169 | 0.9215 |
| 2011-Q1 | 0.8498 | 0.8701 | 0.8994 | 0.9246 | 0.9206 | 0.9093 | 0.9265 | 0.9218 |
| 2011-Q2 | 0.8366 | 0.8618 | 0.8962 | 0.9222 | 0.9187 | 0.9028 | 0.9236 | 0.9182 |
| 2011-Q3 | 0.8397 | 0.8599 | 0.8820 | 0.9114 | 0.9113 | 0.8979 | 0.9168 | 0.9125 |
| 2011-Q4 | 0.8393 | 0.8613 | 0.8773 | 0.9166 | 0.9151 | 0.8987 | 0.9196 | 0.9146 |
| 2012-Q1 | 0.8257 | 0.8589 | 0.8875 | 0.9219 | 0.9221 | 0.9096 | 0.9264 | 0.9234 |
| 2012-Q2 | 0.8304 | 0.8587 | 0.8910 | 0.9243 | 0.9232 | 0.9112 | 0.9275 | 0.9234 |
| 2012-Q3 | 0.8233 | 0.8519 | 0.8841 | 0.9209 | 0.9198 | 0.9048 | 0.9251 | 0.9200 |
| 2012-Q4 | 0.8099 | 0.8389 | 0.8814 | 0.9169 | 0.9168 | 0.9032 | 0.9197 | 0.9130 |
| 2013-Q1 | 0.8057 | 0.8591 | 0.8745 | 0.9253 | 0.9209 | 0.9010 | 0.9256 | 0.9182 |
| 2013-Q2 | 0.8138 | 0.8617 | 0.8839 | 0.9237 | 0.9174 | 0.9052 | 0.9248 | 0.9178 |
| 2013-Q3 | 0.8014 | 0.8516 | 0.8687 | 0.9142 | 0.9096 | 0.8915 | 0.9154 | 0.9086 |
| 2013-Q4 | 0.8201 | 0.8757 | 0.8919 | 0.9271 | 0.9224 | 0.9128 | 0.9288 | 0.9239 |
| 2014-Q1 | 0.8227 | 0.8790 | 0.8954 | 0.9299 | 0.9255 | 0.9190 | 0.9318 | 0.9277 |
| 2014-Q2 | 0.8046 | 0.8628 | 0.8875 | 0.9265 | 0.9238 | 0.9115 | 0.9302 | 0.9249 |
| 2014-Q3 | 0.8108 | 0.8722 | 0.8932 | 0.9333 | 0.9332 | 0.9198 | 0.9392 | 0.9343 |
| 2014-Q4 | 0.8012 | 0.8708 | 0.8916 | 0.9330 | 0.9315 | 0.9190 | 0.9369 | 0.9321 |
| 2015-Q1 | 0.8006 | 0.8878 | 0.8911 | 0.9397 | 0.9366 | 0.9167 | 0.9404 | 0.9362 |
| 2015-Q2 | 0.8054 | 0.8941 | 0.9033 | 0.9452 | 0.9417 | 0.9249 | 0.9470 | 0.9432 |
| 2015-Q3 | 0.8123 | 0.8958 | 0.9084 | 0.9457 | 0.9400 | 0.9274 | 0.9455 | 0.9404 |
| 2015-Q4 | 0.8095 | 0.8960 | 0.9046 | 0.9426 | 0.9392 | 0.9288 | 0.9442 | 0.9399 |
| 2016-Q1 | 0.8158 | 0.8975 | 0.9091 | 0.9489 | 0.9450 | 0.9331 | 0.9515 | 0.9481 |
| 2016-Q2 | 0.8113 | 0.8980 | 0.9038 | 0.9428 | 0.9392 | 0.9279 | 0.9454 | 0.9394 |
| 2016-Q3 | 0.8098 | 0.8959 | 0.9004 | 0.9437 | 0.9400 | 0.9251 | 0.9473 | 0.9424 |
| 2016-Q4 | 0.8106 | 0.9082 | 0.9175 | 0.9540 | 0.9509 | 0.9382 | 0.9566 | 0.9526 |
| 2017-Q1 | 0.8109 | 0.8725 | 0.9163 | 0.9559 | 0.9508 | 0.9384 | 0.9571 | 0.9499 |
| 2017-Q2 | 0.8051 | 0.8776 | 0.9124 | 0.9518 | 0.9483 | 0.9355 | 0.9516 | 0.9483 |
| 2017-Q3 | 0.7913 | 0.8676 | 0.8942 | 0.9432 | 0.9384 | 0.9227 | 0.9426 | 0.9368 |
| 2017-Q4 | 0.7787 | 0.8649 | 0.8941 | 0.9439 | 0.9398 | 0.9229 | 0.9451 | 0.9392 |
| 2018-Q1 | 0.7707 | 0.8686 | 0.8994 | 0.9462 | 0.9407 | 0.9278 | 0.9460 | 0.9403 |
| 2018-Q2 | 0.7388 | 0.8635 | 0.9024 | 0.9470 | 0.9427 | 0.9285 | 0.9491 | 0.9454 |
| 2018-Q3 | 0.7707 | 0.8583 | 0.8977 | 0.9402 | 0.9358 | 0.9218 | 0.9389 | 0.9324 |
| 2018-Q4 | 0.7487 | 0.8455 | 0.9009 | 0.9375 | 0.9330 | 0.9228 | 0.9352 | 0.9290 |

Table A-11: Comparison of different local region sizes (kNNs) using *Euclidean distance*

(LR=Logistic Regression, L=Local classifier, 2k=2000, 4k=4000, 6k=6000 for kNN)

| | 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 |
| 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 | 0.926 | 0.926 | 0.926 | 0.636 | 0.630 | 0.627 |
| StdDev | 0.009 | 0.012 | 0.012 | 0.028 | 0.030 | 0.031 |

Table A-12: Comparison of different local region sizes (kNNs) using Mahalanobis distance

(LR=Logistic Regression, L=Local classifier, 2k=2000, 4k=4000, 6k=6000 for kNN)

| | AUC | | | H-Measure | | |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | LR-L_2k | LR-L_4k | LR-L_6k | LR-L_2k | LR-L_4k | LR-L_6k |
| 2009-Q1 | 0.8982 | 0.9020 | 0.9022 | 0.5714 | 0.5686 | 0.5738 |
| 2009-Q2 | 0.9065 | 0.9129 | 0.9145 | 0.5891 | 0.6045 | 0.6031 |
| 2009-Q3 | 0.9127 | 0.9173 | 0.9183 | 0.5972 | 0.6022 | 0.6004 |
| 2009-Q4 | 0.9032 | 0.9115 | 0.9152 | 0.5790 | 0.5962 | 0.6007 |
| 2010-Q1 | 0.9104 | 0.9185 | 0.9208 | 0.5985 | 0.6107 | 0.6176 |
| 2010-Q2 | 0.8993 | 0.9126 | 0.9201 | 0.5841 | 0.6045 | 0.6116 |
| 2010-Q3 | 0.9024 | 0.9097 | 0.9145 | 0.5737 | 0.5857 | 0.5897 |
| 2010-Q4 | 0.9031 | 0.9122 | 0.9152 | 0.5822 | 0.5958 | 0.5973 |
| 2011-Q1 | 0.8970 | 0.9047 | 0.9062 | 0.5536 | 0.5707 | 0.5709 |
| 2011-Q2 | 0.9002 | 0.9002 | 0.9021 | 0.5668 | 0.5646 | 0.5660 |
| 2011-Q3 | 0.8935 | 0.9025 | 0.9040 | 0.5516 | 0.5683 | 0.5736 |
| 2011-Q4 | 0.8997 | 0.9052 | 0.9081 | 0.5631 | 0.5750 | 0.5783 |
| 2012-Q1 | 0.9029 | 0.9077 | 0.9092 | 0.5671 | 0.5826 | 0.5861 |
| 2012-Q2 | 0.9083 | 0.9131 | 0.9165 | 0.5849 | 0.5942 | 0.6021 |
| 2012-Q3 | 0.8998 | 0.9047 | 0.9085 | 0.5687 | 0.5735 | 0.5828 |
| 2012-Q4 | 0.8957 | 0.9038 | 0.9082 | 0.5596 | 0.5744 | 0.5853 |
| 2013-Q1 | 0.8927 | 0.9011 | 0.9032 | 0.5547 | 0.5624 | 0.5777 |
| 2013-Q2 | 0.8916 | 0.9042 | 0.9097 | 0.5458 | 0.5670 | 0.5835 |
| 2013-Q3 | 0.8813 | 0.8899 | 0.8951 | 0.5265 | 0.5429 | 0.5615 |
| 2013-Q4 | 0.8850 | 0.8981 | 0.9077 | 0.5279 | 0.5529 | 0.5794 |
| 2014-Q1 | 0.8862 | 0.9064 | 0.8963 | 0.5469 | 0.5805 | 0.5639 |
| 2014-Q2 | 0.8843 | 0.8982 | 0.9070 | 0.5273 | 0.5673 | 0.5785 |
| 2014-Q3 | 0.8934 | 0.9075 | 0.9126 | 0.5499 | 0.5817 | 0.5912 |
| 2014-Q4 | 0.8925 | 0.9116 | 0.9187 | 0.5486 | 0.5850 | 0.6033 |
| 2015-Q1 | 0.8789 | 0.9071 | 0.9150 | 0.5340 | 0.5778 | 0.5940 |
| 2015-Q2 | 0.8934 | 0.9108 | 0.9211 | 0.5564 | 0.5886 | 0.6124 |
| 2015-Q3 | 0.8872 | 0.9093 | 0.9200 | 0.5490 | 0.5946 | 0.6144 |
| 2015-Q4 | 0.8550 | 0.8845 | 0.8942 | 0.5202 | 0.5724 | 0.5806 |
| 2016-Q1 | 0.8754 | 0.8975 | 0.9069 | 0.5495 | 0.5789 | 0.5980 |
| 2016-Q2 | 0.8700 | 0.8829 | 0.8922 | 0.5206 | 0.5403 | 0.5437 |
| 2016-Q3 | 0.8617 | 0.8916 | 0.9011 | 0.5234 | 0.5702 | 0.5823 |
| 2016-Q4 | 0.8776 | 0.9014 | 0.9091 | 0.5611 | 0.6152 | 0.6239 |
| 2017-Q1 | 0.8889 | 0.9154 | 0.9249 | 0.5805 | 0.6322 | 0.6462 |
| 2017-Q2 | 0.8989 | 0.9187 | 0.9272 | 0.5817 | 0.6211 | 0.6397 |
| 2017-Q3 | 0.8908 | 0.9018 | 0.9082 | 0.5712 | 0.6118 | 0.6280 |
| 2017-Q4 | 0.8928 | 0.9061 | 0.9191 | 0.5658 | 0.5930 | 0.6181 |
| 2018-Q1 | 0.8923 | 0.8993 | 0.9040 | 0.5650 | 0.5937 | 0.6056 |
| 2018-Q2 | 0.8795 | 0.9077 | 0.9154 | 0.5481 | 0.6026 | 0.6157 |
| 2018-Q3 | 0.8845 | 0.9040 | 0.9156 | 0.5472 | 0.5885 | 0.6028 |
| 2018-Q4 | 0.8781 | 0.9096 | 0.9226 | 0.5614 | 0.6144 | 0.6331 |
| Mean | 0.891 | 0.905 | 0.911 | 0.559 | 0.585 | 0.595 |
| StdDev | 0.013 | 0.008 | 0.009 | 0.021 | 0.021 | 0.023 |

Table A-13: Local vs Global Classifiers (AUC)

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN,
 *= training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot)

| | AUC | | | | | |
|----------|---------------|---------------|---------------|---------------|---------------|---------------|
| | LR-L_2k | XGB-L_2k | RF-L_2k | LR-G | XGB-G | RF-G |
| 2009-Q1* | 0.9100 | 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 | 0.9170 | 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 | 0.8720 | 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 | 0.9240 | 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 | 0.8790 | 0.9299 | 0.9318 |
| 2014-Q2 | 0.9235 | 0.9240 | 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 | 0.9330 | 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 | 0.9470 |
| 2015-Q3 | 0.9317 | 0.9332 | 0.9419 | 0.8958 | 0.9457 | 0.9455 |
| 2015-Q4 | 0.9314 | 0.9307 | 0.9434 | 0.8960 | 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 | 0.8980 | 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 | 0.9540 | 0.9566 |
| 2017-Q1* | 0.9417 | 0.9477 | 0.9580 | 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 | 0.9440 | 0.9517 | 0.8686 | 0.9462 | 0.9460 |
| 2018-Q2 | 0.9351 | 0.9446 | 0.9526 | 0.8635 | 0.9470 | 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 |

Table A-14: Local vs Global Classifiers (H-Measure)

(LR=Logistic Regression, RF=Random Forrest, XGB=Gradient Boosting, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, *= training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot)

| | H-Measure | | | | | |
|----------|---------------|---------------|---------------|---------------|---------------|---------------|
| | 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 |

Table A-15: kNNs vs Random sub-sampling

(LR=Logistic Regression, L=Local classifier, G=Global Classifier, 2k=2000 for kNN, rnd=random,
 *= training snapshot for global classifiers)

| | 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 |

References

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), 39–59.
- Abdou, H. A., & Pointon, J. (2011). Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. *Intelligent Systems in Accounting, Finance and Management*, 18(2–3), 59–88. <https://doi.org/10.1002/isaf.325>
- Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, 73, 1–10. <https://doi.org/10.1016/j.eswa.2016.12.020>
- Adams, N. M., Tasoulis, D. K., Anagnostopoulos, C., & Hand, D. J. (2010). Temporally-adaptive linear classification for handling population drift in credit scoring. *Lechevallier, Y. And Saporta.(Eds), COMPSTAT2010, Proceedings of the 19th International Conference on Computational Statistics*, 167–176.
- Addo, P., Guegan, D., & Hassani, B. (2018). Credit Risk Analysis Using Machine and Deep Learning Models. *Risks*, 6(2), 38. <https://doi.org/10.3390/risks6020038>
- Aggarwal, C. (2014). Instance-based Learning: A Survey. In *Data classification: Algorithms and applications* (p. 29). CRC Press.
- Aggarwal, N. (2021). The norms of algorithmic credit scoring. *The Cambridge Law Journal*, 80(1), 42–73.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66.
- Akerlof, G. A. (1978). The market for “lemons”: Quality uncertainty and the market mechanism. In *Uncertainty in economics* (pp. 235–251). Elsevier.
- Ala’raj, M., & Abbod, M. F. (2016a). Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems*, 104, 89–105. <https://doi.org/10.1016/j.knosys.2016.04.013>
- Ala’raj, M., & Abbod, M. F. (2016b). A new hybrid ensemble credit scoring model based on classifiers consensus system approach. *Expert Systems with Applications*, 64, 36–55. <https://doi.org/10.1016/j.eswa.2016.07.017>
- Albanesi, S., & Vamossy, D. F. (2019). *Predicting Consumer Default: A Deep Learning Approach* (Working Paper No. 26165; Working Paper Series). National Bureau of Economic Research. <https://doi.org/10.3386/w26165>

Ali, A., Shamsuddin, S. M., & Ralescu, A. (2015). *Classification with class imbalance problem: A review*. 7, 176–204.

Allen, L., & Saunders, A. (2002). *A Survey of Cyclical Effects in Credit Risk Measurement Models* (SSRN Scholarly Paper ID 315561). Social Science Research Network. <http://papers.ssrn.com/abstract=315561>

Alonso, A., & Carbó, J. M. (2020). Machine learning in credit risk: Measuring the dilemma between prediction and supervisory cost. In *Working Papers* (No. 2032; Working Papers). Banco de España & Working Papers Homepage. Alonso, Andrés and Carbó, José Manuel, Machine Learning in Credit Risk: Measuring the Dilemma Between Prediction and Supervisory Cost (November 3, 2020). Banco de Espana Working Paper No. 2032, Available at SSRN: <https://ssrn.com/abstract=3724374> or <http://dx.doi.org/10.2139/ssrn.3724374>

Anagnostopoulos, C., Hand, D. J., & Adams, N. M. (2019). *Measuring classification performance: The hmeasure package*. 17.

Anagnostopoulos, C., Tasoulis, D. K., Adams, N. M., & Hand, D. J. (2009). Temporally adaptive estimation of logistic classifiers on data streams. *Advances in Data Analysis and Classification*, 3(3), 243–261. <https://doi.org/10.1007/s11634-009-0051-x>

Anagnostopoulos, C., Tasoulis, D. K., Adams, N. M., Pavlidis, N. G., & Hand, D. J. (2012). Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification. *Statistical Analysis and Data Mining*, 5(2), 139–166. <https://doi.org/10.1002/sam.10151>

Anderson, R. (2007). *The credit scoring toolkit: Theory and practice for retail credit risk management and decision automation*. Oxford University Press.

Anderson, R. (2019). *Credit Intelligence & Modelling: Many Paths through the Forest*. Independently published.

Anderson, R. A. (2022). *Credit Intelligence & Modelling: Many Paths through the Forest of Credit Rating and Scoring*. Oxford University Press.

Andreeva, G., Ansell, J., & Crook, J. (2007). Modelling profitability using survival combination scores. *European Journal of Operational Research*, 183(3), 1537–1549. <https://doi.org/10.1016/j.ejor.2006.10.064>

Ashcraft, A. B., & Schuermann, T. (2008). *Understanding the Securitization of Subprime Mortgage Credit* (SSRN Scholarly Paper ID 1071189). Social Science Research Network. <http://papers.ssrn.com/abstract=1071189>

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*(1–5), 11–73.

Avery, R. B., Bostic, R. W., Calem, P. S., & Canner, G. B. (2000). Credit Scoring: Statistical Issues and Evidence from Credit-Bureau Files. *Real Estate Economics*, *28*(3), 523–547. <https://doi.org/10.1111/1540-6229.00811>

Bag, P., & Jacobs, M. (2012). Parsimonious exposure-at-default modeling for unfunded loan commitments. *The Journal of Risk Finance*.

Bank of England. (2019). *Machine learning in UK financial services* (p. 36). Bank of England.

Barddal, J. P., Loezer, L., Enembreck, F., & Lanzaolo, R. (2020). Lessons learned from data stream classification applied to credit scoring. *Expert Systems with Applications*, 113899. <https://doi.org/10.1016/j.eswa.2020.113899>

Bellotti, T., & Crook, J. (2008). Credit scoring with macroeconomic variables using survival analysis. *Journal of the Operational Research Society*, *60*(12), 1699–1707.

Bellotti, T., & Crook, J. (2012). Loss given default models incorporating macroeconomic variables for credit cards. *International Journal of Forecasting*, *28*(1), 171–182. <https://doi.org/10.1016/j.ijforecast.2010.08.005>

Bellotti, T., & Crook, J. (2014). Retail credit stress testing using a discrete hazard model with macroeconomic factors. *The Journal of the Operational Research Society*, *65*(3), 340–350. <http://dx.doi.org/10.1057/jors.2013.91>

Bennardo, A., Pagano, M., & Piccolo, S. (2015). Multiple Bank Lending, Creditor Rights, and Information Sharing*. *Review of Finance*, *19*(2), 519–570. <https://doi.org/10.1093/rof/rfu001>

Bequé, A., & Lessmann, S. (2017). Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications*, *86*, 42–53. <https://doi.org/10.1016/j.eswa.2017.05.050>

Bergmann, B., & Hommel, G. (1988). Improvements of General Multiple Test Procedures for Redundant Systems of Hypotheses. In P. Bauer, G. Hommel, & E. Sonnemann (Eds.), *Multiple Hypothesenprüfung / Multiple Hypotheses Testing* (pp. 100–115). Springer. https://doi.org/10.1007/978-3-642-52307-6_8

Bhatore, S., Mohan, L., & Reddy, Y. R. (2020). Machine learning techniques for credit risk evaluation: A systematic literature review. *Journal of Banking and Financial Technology*. <https://doi.org/10.1007/s42786-020-00020-3>

Bifet, A., Gama, J., Pechenizkiy, M., & Žliobaitė, I. (2011). *Tutorial: Handling Concept Drift: Importance, Challenges and Solutions* [Conference Tutorial]. PAKDD.

Bijak, K. (2011). Kalman filtering as a performance monitoring technique for a propensity scorecard. *The Journal of the Operational Research Society*, 62(1), 29–37. <http://dx.doi.org/10.1057/jors.2009.183>

Bijak, K., & Thomas, L. C. (2012). Does segmentation always improve model performance in credit scoring? *Expert Syst. Appl.*, 39(3), 2433–2442. <https://doi.org/10.1016/j.eswa.2011.08.093>

Bischl, B., Kühn, T., & Szepannek, G. (2016). On Class Imbalance Correction for Classification Algorithms in Credit Scoring. In M. Lübbecke, A. Koster, P. Letmathe, R. Madlener, B. Peis, & G. Walther (Eds.), *Operations Research Proceedings 2014* (pp. 37–43). Springer International Publishing. https://doi.org/10.1007/978-3-319-28697-6_6

Bonfim, D. (2009). Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics. *Journal of Banking & Finance*, 33(2), 281–299.

Bontempi, G., Bersini, H., & Birattari, M. (2001). The local paradigm for modeling and control: From neuro-fuzzy to lazy learning. *Fuzzy Sets and Systems*, 121(1), 59–72. [https://doi.org/10.1016/S0165-0114\(99\)00172-4](https://doi.org/10.1016/S0165-0114(99)00172-4)

Bontempi, G., Birattari, M., & Bersini, H. (2002). Lazy learning: A logical method for supervised learning. In *New learning paradigms in soft computing* (pp. 97–136). Springer. http://link.springer.com/chapter/10.1007/978-3-7908-1803-1_4

Bottou, L., & Vapnik, V. (1992). Local learning algorithms. *Neural Computation*, 4(6), 888–900.

Boyacioglu, M. A., Kara, Y., & Baykan, Ö. K. (2009). Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey. *Expert Systems with Applications*, 36(2, Part 2), 3355–3366. <https://doi.org/10.1016/j.eswa.2008.01.003>

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2), 1–50.

Breeden, J. (2014). *Reinventing Retail Lending Analytics—2nd Impression*. Risk Books.

Breeden, J. L., Parker, R., & Steinebach, C. (2012). A through-the-cycle model for retail lending economic capital. *International Journal of Forecasting*, 28(1), 133–138. <https://doi.org/10.1016/j.ijforecast.2011.01.005>

Breeden, J. L., & Thomas, L. C. (2008). The relationship between default and economic cycles for retail portfolios across countries. *Journal of Risk Model Validation*, 2(3), 11–47.

Breeden, J., Thomas, L., & McDonald III, J. (2007). Stress testing retail load portfolios with dual-time dynamics. *Journal of Risk Model Validation*, 2(2), 1–19.

Britto, A. S., Sabourin, R., & Oliveira, L. E. S. (2014). Dynamic selection of classifiers—A comprehensive review. *Pattern Recognition*, 47(11), 3665–3680. <https://doi.org/10.1016/j.patcog.2014.05.003>

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3), 3446–3453. <https://doi.org/10.1016/j.eswa.2011.09.033>

Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2020). Explainable AI in Fintech Risk Management. *Frontiers in Artificial Intelligence*, 3. <https://doi.org/10.3389/frai.2020.00026>

Calabrese, R. (2014). Downturn Loss Given Default: Mixture distribution estimation. *European Journal of Operational Research*, 237(1), 271–277. <https://doi.org/10.1016/j.ejor.2014.01.043>

Camacho, L., Douzas, G., & Bacao, F. (2022). Geometric SMOTE for regression. *Expert Systems with Applications*, 116387.

Chandler, G. G., & Coffman, J. Y. (1979). A Comparative Analysis of Empirical Vs. Judgmental Credit Evaluation. *Financial Review*, 14(4), 23–23. <https://doi.org/10.1111/j.1540-6288.1979.tb01773.x>

Chang, Y.-C., Chang, K.-H., & Wu, G.-J. (2018). Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing*, 73, 914–920. <https://doi.org/10.1016/j.asoc.2018.09.029>

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>

Chen, F.-L., & Li, F.-C. (2010). Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37(7), 4902–4909. <https://doi.org/10.1016/j.eswa.2009.12.025>

Chen, Y. (2012). Research on Multi-Classification of Credit Rating of Small and Medium-Sized Enterprises in Growth Enterprises Board Based on Fuzzy Ordinal Regression Support Vector Machine. *International Journal of Economics & Finance*, 4(3), 248–252. <https://doi.org/10.5539/ijef.v4n3p248>

Cleveland, W. S., Devlin, S. J., & Grosse, E. (1988). Regression by local fitting: Methods, properties, and computational algorithms. *Journal of Econometrics*, 37(1), 87–114.

Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, 28(1), 224–238. <https://doi.org/10.1016/j.ijforecast.2011.07.006>

Crook, J., & Bellotti, T. (2010). Time varying and dynamic models for default risk in consumer loans. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(2), 283–305. <https://doi.org/10.1111/j.1467-985X.2009.00617.x>

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183(3), 1447–1465.

Crook, J. N., Thomas, L. C., & Hamilton, R. (1992). The degradation of the scorecard over the business cycle. *IMA Journal of Management Mathematics*, 4(1), 111–123.

Cruz, R. M. O., Cavalcanti, G. D. C., & Ren, T. I. (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. <https://doi.org/10.1109/IJCNN.2011.6033350>

Cruz, R. M. O., Sabourin, R., & Cavalcanti, G. D. C. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41, 195–216. <https://doi.org/10.1016/j.inffus.2017.09.010>

Cruz, R. M. O., Zakane, H. H., Sabourin, R., & Cavalcanti, G. D. C. (2017). Dynamic Ensemble Selection VS K-NN: Why and when Dynamic Selection obtains higher classification performance? *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–6. <https://doi.org/10.1109/IPTA.2017.8310100>

Danenas, P., & Garsva, G. (2012). Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach. *Procedia Computer Science*, 9, 1324–1333. <https://doi.org/10.1016/j.procs.2012.04.145>

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91, 106263. <https://doi.org/10.1016/j.asoc.2020.106263>

Daumé III, H., & Marcu, D. (2006). Domain Adaptation for Statistical Classifiers. *J. Artif. Intell. Res.(JAIR)*, 26, 101–126.

de Haro-García, A., Cerruela-García, G., & García-Pedrajas, N. (2019). Instance selection based on boosting for instance-based learners. *Pattern Recognition*, 96, 106959. <https://doi.org/10.1016/j.patcog.2019.07.004>

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1–30.

Demyanyk, Y., & Van Hemert, O. (2008). *Understanding the Subprime Mortgage Crisis* (SSRN Scholarly Paper ID 1020396). Social Science Research Network. <https://doi.org/10.2139/ssrn.1020396>

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1–15.

Dirick, L., Claeskens, G., & Baesens, B. (2015). Time to default in credit scoring using survival analysis: A benchmark study. Available at SSRN 2663267. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2663267

Djeundje, V. B., & Crook, J. (2018). Incorporating heterogeneity and macroeconomic variables into multi-state delinquency models for credit cards. *European Journal of Operational Research*, 271(2), 697–709. <https://doi.org/10.1016/j.ejor.2018.05.040>

Domeniconi, C., Gunopulos, D., & others. (2001). Adaptive nearest neighbor classification using support vector machines. *NIPS*, 665–672. <https://papers.nips.cc/paper/2054-adaptive-nearest-neighbor-classification-using-support-vector-machines.pdf>

Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9), 1281–1285. <https://doi.org/10.1109/TPAMI.2002.1033219>

Douzas, G., & Bacao, F. (2017). Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert Systems with Applications*, 82, 40–52.

Douzas, G., & Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications*, 91, 464–471.

Douzas, G., Rauch, R., & Bacao, F. (2021). G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE. *Expert Systems with Applications*, 183, 115230.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification* (2nd edition). Wiley-Interscience.

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297(3), 1178–1192.

Durand, D. (1941). Credit-Rating Formulae. In *Risk Elements in Consumer Installment Financing* (pp. 83–91). NBER. <http://www.nber.org/chapters/c9265.pdf>

EBA. (2020). *Final Report on Big Data and Advanced Analytics*. European Banking Authority.

Elliott, R. J., & Filinkov, A. (2008). A self tuning model for risk estimation. *Expert Systems with Applications*, 34(3), 1692–1697. <https://doi.org/10.1016/j.eswa.2007.01.044>

Fahner, G. (2012). Estimating causal effects of credit decisions. *International Journal of Forecasting*, 28(1), 248–260. <https://doi.org/10.1016/j.ijforecast.2010.10.002>

Feng, X., Xiao, Z., Zhong, B., Qiu, J., & Dong, Y. (2018). Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing*, 65, 139–151. <https://doi.org/10.1016/j.asoc.2018.01.021>

Figlewski, S., Frydman, H., & Liang, W. (2012). Modeling the effect of macroeconomic factors on corporate default and credit rating transitions. *International Review of Economics & Finance*, 21(1), 87–105.

Finlay, S. (2010). Credit scoring for profitability objectives. *European Journal of Operational Research*, 202(2), 528–537. <https://doi.org/10.1016/j.ejor.2009.05.025>

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179–188.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Advances in Artificial Intelligence—SBIA 2004* (pp. 286–295). Springer. http://link.springer.com/chapter/10.1007/978-3-540-28645-5_29

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Comput. Surv.*, 46(4), 44:1-44:37. <https://doi.org/10.1145/2523813>

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4), 42–47.

Gao, J., Fan, W., Han, J., & Yu, P. S. (2007). A general framework for mining concept-drifting data streams with skewed distributions. In *Proc. SDM'07*.

García, S., Derrac, J., Cano, J. R., & Herrera, F. (2012). Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), 417–435.

<https://doi.org/10.1109/TPAMI.2011.142>

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. *Information Sciences*, 180(10), 2044–2064.

<https://doi.org/10.1016/j.ins.2009.12.010>

García, V., Marqués, A. I., & Sánchez, J. S. (2012). Improving Risk Predictions by Preprocessing Imbalanced Credit Data. In T. Huang, Z. Zeng, C. Li, & C. S. Leung (Eds.), *Neural Information Processing* (Vol. 7664, pp. 68–75). Springer Berlin Heidelberg.

https://doi.org/10.1007/978-3-642-34481-7_9

García, V., Marqués, A. I., & Sánchez, J. S. (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. <https://doi.org/10.1016/j.inffus.2018.07.004>

García, V., Sánchez, J. S., Ochoa-Ortiz, A., & López-Najera, A. (2019). Instance Selection for the Nearest Neighbor Classifier: Connecting the Performance to the Underlying Data Structure. In A. Morales, J. Fierrez, J. S. Sánchez, & B. Ribeiro (Eds.), *Pattern Recognition and Image Analysis* (pp. 249–256). Springer International Publishing.

https://doi.org/10.1007/978-3-030-31332-6_22

García, S., & Herrera, F. (2008). An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9, 18.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 80–89.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining Explanations: An Overview of Interpretability of Machine Learning. *ArXiv:1806.00069 [Cs, Stat]*. <http://arxiv.org/abs/1806.00069>

Grimshaw, S. D., & Alexander, W. P. (2011). Markov chain models for delinquency: Transition matrix estimation and forecasting. *Applied Stochastic Models in Business & Industry*, 27(3), 267–279. <https://doi.org/10.1002/asmb.827>

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. <https://doi.org/10.1016/j.jfds.2018.04.001>

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.

Gül, S., Kabak, Ö., & Topcu, I. (2018). A multiple criteria credit rating approach utilizing social media data. *Data & Knowledge Engineering*, 116, 80–99. <https://doi.org/10.1016/j.datak.2018.05.005>

Gunnarsson, B. R., Broucke, S. vanden, Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep Learning for Credit Scoring: Do or Don't? *European Journal of Operational Research*, S037722172100196X. <https://doi.org/10.1016/j.ejor.2021.03.006>

Guo, S., He, H., & Huang, X. (2019). A Multi-Stage Self-Adaptive Classifier Ensemble Model With Application in Credit Scoring. *IEEE Access*, 7, 78549–78559. <https://doi.org/10.1109/ACCESS.2019.2922676>

Guo, Y., Zhou, W., Luo, C., Liu, C., & Xiong, H. (2016). Instance-based credit risk assessment for investment decisions in P2P lending. *European Journal of Operational Research*, 249(2), 417–426. <https://doi.org/10.1016/j.ejor.2015.05.050>

Gürtler, M., Hibbeln, M. T., & Usselman, P. (2018). Exposure at default modeling—A theoretical and empirical assessment of estimation approaches and parameter choice. *Journal of Banking & Finance*, 91, 176–188.

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. <https://doi.org/10.3390/jrfm11010012>

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *Advances in Intelligent Computing* (Vol. 3644, pp. 878–887). Springer Berlin Heidelberg. https://doi.org/10.1007/11538059_91

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1), 103–123.

Hand, D. J., & Anagnostopoulos, C. (2013). When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? *Pattern Recognition Letters*, 34(5), 492–495. <https://doi.org/10.1016/j.patrec.2012.12.004>

Hand, D. J., & Anagnostopoulos, C. (2021). Notes on the H-measure of classifier performance. *ArXiv:2106.11888 [Cs]*. <http://arxiv.org/abs/2106.11888>

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *160*(3), 523–541.

Hardt, M., Price, E., & Srebro, N. (2016a). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, *29*.

Hardt, M., Price, E., & Srebro, N. (2016b). Equality of Opportunity in Supervised Learning. *ArXiv:1610.02413 [Cs]*. <http://arxiv.org/abs/1610.02413>

Harris, T. (2015). Credit scoring using the clustered support vector machine. *Expert Systems with Applications*, *42*(2), 741–750. <https://doi.org/10.1016/j.eswa.2014.08.029>

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, *98*, 105–117. <https://doi.org/10.1016/j.eswa.2018.01.012>

Hibbeln, M., Norden, L., Usselman, P., & Gürtler, M. (2019). Informational synergies in consumer credit. *Journal of Financial Intermediation*, 100831. <https://doi.org/10.1016/j.jfi.2019.100831>

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hsieh, H.-I., Lee, T.-P., & Lee, T.-S. (2010). Data Mining in Building Behavioral Scoring Models. *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference On*, 1–4. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5677005

Hurley, M., & Adebayo, J. (2016). Credit scoring in the era of big data. *Yale JL & Tech.*, *18*, 148.

Hurlin, C., Pérignon, C., & Saurin, S. (2021). *The Fairness of Credit Scoring Models* (SSRN Scholarly Paper ID 3785882). Social Science Research Network. <https://doi.org/10.2139/ssrn.3785882>

ICCR. (2018). *Use of Alternative Data to Enhance Credit Reporting to Enable Access to Digital Financial Services by Individuals and SMEs operating in the Informal Economy*. INTERNATIONAL COMMITTEE ON CREDIT REPORTING (ICCR). https://www.gpfi.org/sites/gpfi/files/documents/Use_of_Alternative_Data_to_Enhance_Credit_Reporting_to_Enable_Access_to_Digital_Financial_Services_ICCR.pdf

- ICCR. (2019). *Credit Scoring Approaches Guidelines*. World Bank. <https://doi.org/10.1596/31806>
- Im, J. -k, Apley, D. W., Qi, C., & Shan, X. (2012). A time-dependent proportional hazards survival model for credit risk analysis. *The Journal of the Operational Research Society*, *63*(3), 306–321. <http://dx.doi.org.libezproxy.open.ac.uk/10.1057/jors.2011.34>
- Institute of International Finance. (2019). *Machine learning in credit risk*.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, *3*(1), 79–87. <https://doi.org/10.1162/neco.1991.3.1.79>
- Jamain, A., & Hand, D. J. (2009). Where are the large and difficult datasets? *Advances in Data Analysis and Classification*, *3*(1), 25–38. <https://doi.org/10.1007/s11634-009-0037-8>
- Japkowicz, N., & Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.
- Jo, H., Han, I., & Lee, H. (1997). Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. *Expert Systems with Applications*, *13*(2), 97–108.
- Jung, K. M., Thomas, L. C., & So, M. C. (2015). When to rebuild or when to adjust scorecards. *Journal of the Operational Research Society*, *66*(10), 1656–1668. <https://doi.org/10.1057/jors.2015.43>
- Kaposty, F., Löderbusch, M., & Maciag, J. (2017). Stochastic loss given default and exposure at default in a structural model of portfolio credit risk. *Journal of Credit Risk*, *13*(1).
- Kaufman, R. (2018, August 21). *The History of the FICO® Score*. <https://www.myfico.com/credit-education/blog/history-of-the-fico-score>
- Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Computing Surveys*, *52*(4), 1–36. <https://doi.org/10.1145/3343440>
- Kelly, M. G., Hand, D. J., & Adams, N. M. (1999). The impact of changing populations on classifier performance. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 367–371.
- Kennedy, K., Mac Namee, B., Delany, S. J., O’Sullivan, M., & Watson, N. (2013). A window of opportunity: Assessing behavioural scoring. *Expert Systems with Applications*, *40*(4), 1372–1380. <https://doi.org/10.1016/j.eswa.2012.08.052>

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*.

Klinkenberg, R. (2004). Learning Drifting Concepts: Example Selection vs. Example Weighting. *Intell. Data Anal.*, 8(3), 281–300.

Klinkenberg, R., & Joachims, T. (2000). Detecting concept drift with support vector machines. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 11.

Kozodoi, N., Jacob, J., & Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, 297(3), 1083–1094. <https://doi.org/10.1016/j.ejor.2021.06.023>

Kuncheva, L. I. (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 vol.1. <https://doi.org/10.1109/KES.2000.885788>

Kuncheva, L. I. (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. [zotero://attachment/1038/](https://doi.org/10.1007/978-3-540-24143-7_1)

Kuncheva, L. I. (2008). Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. *Proceedings of the 2nd Workshop SUEMA, 2008*, 5–10. <http://www.bangor.ac.uk/~mas00a/papers/lkSUEMA2008.pdf>

Kuncheva, L. I., Arnaiz-González, Á., Díez-Pastor, J.-F., & Gunn, I. A. D. (2019). Instance selection improves geometric mean accuracy: A study on imbalanced data classification. *Progress in Artificial Intelligence*, 8(2), 215–228. <https://doi.org/10.1007/s13748-019-00172-4>

Kvamme, H., Sellereite, N., Aas, K., & Sjørusen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, 102, 207–217. <https://doi.org/10.1016/j.eswa.2018.02.029>

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. https://doi.org/10.1007/978-3-319-05458-2_11

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.
- Leow, M., & Crook, J. (2014). Intensity models and transition probabilities for credit card loan delinquencies. *European Journal of Operational Research*, *236*(2), 685–694. <https://doi.org/10.1016/j.ejor.2013.12.026>
- Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136. <https://doi.org/10.1016/j.ejor.2015.05.030>
- Lewis, E. M. (1992). *An Introduction to Credit Scoring* (2nd ed edition). Fair, Isaac and Co.
- 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 Recognition*, *48*(4), 1523–1537. <https://doi.org/10.1016/j.patcog.2014.10.001>
- Li, F.-C. (2009). The Hybrid Credit Scoring Strategies Based on KNN Classifier. *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, 330–334. <https://doi.org/10.1109/FSKD.2009.261>
- Li, K., Zhou, F., Li, Z., Yao, X., & Zhang, Y. (2021). Predicting loss given default using post-default information. *Knowledge-Based Systems*, *224*, 107068. <https://doi.org/10.1016/j.knosys.2021.107068>
- 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. <https://doi.org/10.1145/3437963.3441743>
- Lim, M. K., & Sohn, S. Y. (2007). Cluster-based dynamic scoring model. *Expert Systems with Applications*, *32*(2), 427–431. <https://doi.org/10.1016/j.eswa.2005.12.006>
- Liu, F., Hua, Z., & Lim, A. (2015). Identifying future defaulters: A hierarchical Bayesian method. *European Journal of Operational Research*, *241*(1), 202–211. <https://doi.org/10.1016/j.ejor.2014.08.008>
- Liu, Z., & Pan, S. (2018). Fuzzy-Rough Instance Selection Combined with Effective Classifiers in Credit Scoring. *Neural Processing Letters*, *47*(1), 193–202. <https://doi.org/10.1007/s11063-017-9641-3>
- Loader, C. (1999). *Local regression and likelihood*. Springer Science & Business Media. <https://books.google.com/books?hl=en&lr=&id=NpjeBwAAQBAJ&oi=fnd&pg=PA1&d>

q=%22so+the+reader+can+pick+those+of+most%22+%22direct+practical+relevance.+F
or+example,+theoretical+motivation%22+%22with+the+theory,+we+also+attempt+to+in
troduce+understanding+of%22+&ots=wVbc1ZlQ76&sig=Usqw8cVzbY-MuivblS-
Av4pwjec

Loterman, G., Brown, I., Martens, D., Mues, C., & Baesens, B. (2012). Benchmarking regression algorithms for loss given default modeling. *International Journal of Forecasting*, 28(1), 161–170. <https://doi.org/10.1016/j.ijforecast.2011.01.006>

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, 65, 465–470. <https://doi.org/10.1016/j.engappai.2016.12.002>

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 Recognition*, 91, 216–231. <https://doi.org/10.1016/j.patcog.2019.02.023>

Malik, M., & Thomas, L. C. (2012). Transition matrix models of consumer credit ratings. *International Journal of Forecasting*, 28(1), 261–272. <https://doi.org/10.1016/j.ijforecast.2011.01.007>

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 [Cs, Stat]*. <http://arxiv.org/abs/1907.12363>

Marqués, A. I., García, V., & Sánchez, J. S. (2012). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, 64(7), 1060–1070. <https://doi.org/10.1057/jors.2012.120>

Masoudnia, S., & Ebrahimpour, R. (2014). Mixture of experts: A literature survey. *Artificial Intelligence Review*, 42(2), 275–293. <https://doi.org/10.1007/s10462-012-9338-y>

Mays, E. (2005). *Handbook of credit scoring*. Publishers Group Uk.

McIntosh, C., & Wydick, B. (2009). What Do Credit Bureaus Do? Understanding Screening, Incentive, and Credit Expansion Effects. *Economics*, 3. <https://repository.usfca.edu/econ/3>

Melo Junior, L., Nardini, F. M., Renso, C., Trani, R., & Macedo, J. A. (2020). A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. *Expert Systems with Applications*, 152, 113351. <https://doi.org/10.1016/j.eswa.2020.113351>

Michael Turner, Robin Varghese, & Patrick Walker. (2015). *Research Consensus Confirms Benefits of Alternative Data* [Technical Report]. PERC.

More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets. *ArXiv:1608.06048 [Cs, Stat]*.
<http://arxiv.org/abs/1608.06048>

Morini, M. (2011). *Understanding and Managing Model Risk: A practical guide for quants, traders and validators*. John Wiley & Sons.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications*, 9(1), 141–142.

Niklis, D., Doumpos, M., & Zopounidis, C. (2014). Combining market and accounting-based models for credit scoring using a classification scheme based on support vector machines. *Applied Mathematics and Computation*, 234, 69–81.
<https://doi.org/10.1016/j.amc.2014.02.028>

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, 34(2), 133–143.
<https://doi.org/10.1007/s10462-010-9165-y>

Óskarsdóttir, M., Bravo, C., Sarraute, C., Vanthienen, J., & Baesens, B. (2019). The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics. *Applied Soft Computing*, 74, 26–39.
<https://doi.org/10.1016/j.asoc.2018.10.004>

Padilla, A. J., & Pagano, M. (1997). Endogenous Communication Among Lenders and Entrepreneurial Incentives. *The Review of Financial Studies*, 10(1), 205–236.
<https://doi.org/10.1093/rfs/10.1.205>

Padilla, A. J., & Pagano, M. (2000). Sharing default information as a borrower discipline device. *European Economic Review*, 44(10), 1951–1980. [https://doi.org/10.1016/S0014-2921\(00\)00055-6](https://doi.org/10.1016/S0014-2921(00)00055-6)

Paleologo, G., Elisseeff, A., & Antonini, G. (2010). Subbagging for credit scoring models. *European Journal Of Operational Research*, 201(2), 490–499.
<https://doi.org/10.1016/j.ejor.2009.03.008>

Papouskova, M., & Hajek, P. (2019). Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decision Support Systems*, 118, 33–45.
<https://doi.org/10.1016/j.dss.2019.01.002>

- Parker, C. (2011). An Analysis of Performance Measures for Binary Classifiers. *2011 IEEE 11th International Conference on Data Mining*, 517–526. <https://doi.org/10.1109/ICDM.2011.21>
- Pavlidis, N. G., Tasoulis, D. K., Adams, N. M., & Hand, D. J. (2011). λ -perceptron: An adaptive classifier for data streams. *Pattern Recognition*, 44(1), 78–96. <https://doi.org/10.1016/j.patcog.2010.07.026>
- Pavlidis, N. G., Tasoulis, D. K., Adams, N. M., & Hand, D. J. (2012). Adaptive consumer credit classification. *Journal of the Operational Research Society*, 63(12), 1645–1654. <https://doi.org/10.1057/jors.2012.15>
- Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *The Journal of Machine Learning Research*, 4, 211–255.
- 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. *FC Bulletins Chapters, in: Bank for International Settlements (Ed.), The Use of Big Data Analytics and Artificial Intelligence in Central Banking*, 50. <https://www.semanticscholar.org/paper/A-robust-machine-learning-approach-for-credit-risk-Petropoulos-Siakoulis/cbae059d97bf674e02d391f939297b31319032ec>
- Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *ArXiv Preprint ArXiv:1009.6119*. <http://arxiv.org/abs/1009.6119>
- Ping, Y., & Yongheng, L. (2011). Neighborhood rough set and SVM based hybrid credit scoring classifier. *Expert Systems with Applications*, 38(9), 11300–11304. <https://doi.org/10.1016/j.eswa.2011.02.179>
- Qazi, N., & Raza, K. (2012). *Effect of Feature Selection, SMOTE and under Sampling on Class Imbalance Classification*. 145–150. <https://doi.org/10.1109/UKSim.2012.116>
- Rahman, M. M., & Davis, D. N. (2013). Addressing the Class Imbalance Problem in Medical Datasets. *International Journal of Machine Learning and Computing*, 224–228. <https://doi.org/10.7763/IJMLC.2013.V3.307>
- Rezac, M., & Rezac, F. (2011). How to Measure the Quality of Credit Scoring Models. *Czech Journal of Economics and Finance (Finance a Uver)*, 61(5), 486–507.

- 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. https://www.schufa4b.de/media/themenundprojekte/downloads_6/studieaufenglisch.pdf
- Saha, A., & Siddiqi, N. (2011, August). *Survival Analysis Workflow: Assessing the impact of Macro-Economic Shocks on Credit Portfolios and Predicting the Time of Default*.
- Saia, R., Carta, S., & Fenu, G. (2018). A wavelet-based data analysis to credit scoring. *Proceedings of the 2nd International Conference on Digital Signal Processing*, 176–180.
- Santafe, G., Inza, I., & Lozano, J. A. (2015). Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, 44(4), 467–508. <https://doi.org/10.1007/s10462-015-9433-y>
- Sarlija, N., Bencic, M., & Zekic-Susac, M. (2006). A neural network classification of credit applicants in consumer credit scoring. *Proceedings of the 24th IASTED International Conference on Artificial Intelligence and Applications*, 205–210. <http://dl.acm.org/citation.cfm?id=1166890.1166925>
- Sarmanova, A., & Albayrak, S. (2013). *Alleviating the Class Imbalance problem in Data Mining*. 6.
- Schaal, S., & Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8), 2047–2084.
- Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, 13(1), 135–143. <https://doi.org/10.1007/BF00993106>
- Schwarz, A., & Arminger, G. (2005). Credit Scoring Using Global and Local Statistical Models. In C. Weihs & W. Gaul (Eds.), *Classification—The Ubiquitous Challenge* (pp. 442–449). Springer Berlin Heidelberg.
- Siddiqi, N. (2005). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley.
- Sigrist, F., & Hirnschall, C. (2019). Grabit: Gradient tree-boosted Tobit models for default prediction. *Journal of Banking & Finance*, 102, 177–192. <https://doi.org/10.1016/j.jbankfin.2019.03.004>
- Siham, A., Sara, S., & Abdellah, A. (2021). Feature selection based on machine learning for credit scoring: An evaluation of filter and embedded methods. *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6.

- Sirignano, J., & Cont, R. (2018). Universal features of price formation in financial markets: Perspectives from Deep Learning. *ArXiv:1803.06917 [q-Fin, Stat]*. <http://arxiv.org/abs/1803.06917>
- Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9), 1449–1459.
- 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
- Soares, R. G. F., Santana, A., Canuto, A. M. P., & de Souto, M. C. P. (2006). Using Accuracy and Diversity to Select Classifiers to Build Ensembles. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 1310–1316. <https://doi.org/10.1109/IJCNN.2006.246844>
- Sousa, M. R., Gama, J., & Brandão, E. (2013). *Introducing time-changing economics into credit scoring* (No. 513; FEP Working Papers). Universidade do Porto, Faculdade de Economia do Porto. <http://www.fep.up.pt/investigacao/workingpapers/wp513.pdf>
- Sousa, M. R., Gama, J., & Brandão, E. (2016). A new dynamic modeling framework for credit risk assessment. *Expert Systems with Applications*, 45, 341–351. <https://doi.org/10.1016/j.eswa.2015.09.055>
- Sousa, M. R., Gama, J., & Gonçalves, M. J. S. (2014). A two-stage model for dealing with temporal degradation of credit scoring. *ArXiv:1406.7775 [q-Fin]*. <http://arxiv.org/abs/1406.7775>
- Stelzer, A. (2019). Predicting credit default probabilities using machine learning techniques in the face of unequal class distributions. *ArXiv:1907.12996 [Cs, Econ]*. <http://arxiv.org/abs/1907.12996>
- Stiefmueller, C. M. (2020). Open Banking and PSD 2: The Promise of Transforming Banking by ‘Empowering Customers.’ In J. Spohrer & C. Leitner (Eds.), *Advances in the Human Side of Service Engineering* (pp. 299–305). Springer International Publishing. https://doi.org/10.1007/978-3-030-51057-2_41
- Stiglitz, J. E., & Weiss, A. (1981). Credit Rationing in Markets with Imperfect Information. *The American Economic Review*, 71(3), 393–410.
- Sun, J., & Li, H. (2011). Dynamic financial distress prediction using instance selection for the disposal of concept drift. *Expert Systems with Applications*, 38(3), 2566–2576. <https://doi.org/10.1016/j.eswa.2010.08.046>

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04), 687–719. <https://doi.org/10.1142/S0218001409007326>

Suresh, H., & Guttag, J. V. (2019). A framework for understanding unintended consequences of machine learning. *ArXiv Preprint ArXiv:1901.10002*, 2(8).

Takada, H., & Sumita, U. (2011). Credit risk model with contagious default dependencies affected by macro-economic condition. *European Journal of Operational Research*, 214(2), 365–379. <https://doi.org/10.1016/j.ejor.2011.05.001>

Tang, B., & Qiu, S. B. (2012). Multi-Class Support Vector Machine for Credit Scoring. *Applied Mechanics and Materials*, 235. <http://dx.doi.org/10.4028/www.scientific.net/AMM.235.419>

Thomas. (2007, August). *Measuring the Discrimination Quality of Suites of Scorecards: ROCs, Gini's, Bounds and Segmentation*. Credit Scoring and Credit Control X, Edinburgh.

Thomas, L. C. (2003). Consumer credit modelling: Context and current issues. *Workshop Paper Presented on the Banff Credit Risk Conference*. https://archytas.birs.ca/workshops/2003/03w5023/files/Thomas_Consumer.pdf

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit Scoring & Its Applications (Monographs on Mathematical Modeling and Computation)* (1st edition). Society for Industrial and Applied Mathematics.

Thomas, L. C., & Malik, M. (2010). Comparison of credit risk models for portfolios of retail loans based on behavioural scores. In D. Rausch & H. Scheule (Eds.), *Model Risk in Financial Crises* (pp. 209–232). Risk Books. <http://eprints.soton.ac.uk/71278/>

Thomas, L. C., Oliver, R. W., & Hand, D. J. (2005). A survey of the issues in consumer credit modelling research. *Journal of the Operational Research Society*, 56(9), 1006–1015. <https://doi.org/10.1057/palgrave.jors.2602018>

Titsias, M. K., & Likas, A. (2002). Mixture of Experts Classification Using a Hierarchical Mixture Model. *Neural Computation*, 14(9), 2221–2244. <https://doi.org/10.1162/089976602320264060>

Tobback, E., & Martens, D. (2019). Retail credit scoring using fine-grained payment data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, rssa.12469. <https://doi.org/10.1111/rssa.12469>

Tobback, E., Martens, D., Van Gestel, T., & Baesens, B. (2014). Forecasting Loss Given Default models: Impact of account characteristics and the macroeconomic state. *Journal*

- Adaptive Credit Scoring using Local Classification Methods of the Operational Research Society, 65(3), 376–392. 133
<https://doi.org/10.1057/jors.2013.158>
- Tobias Berg, Valentin Burg, Ana Gombović, & Manju Puri. (2018). *On the Rise of Fintechs – Credit Scoring Using Digital Footprints* (Working Paper Working Paper 24551; NBER WORKING PAPER SERIES). NATIONAL BUREAU OF ECONOMIC RESEARCH. <http://www.nber.org/papers/w24551>
- Tomczak, J. M., & Zięba, M. (2015). Classification Restricted Boltzmann Machine for comprehensible credit scoring model. *Expert Systems with Applications*, 42(4), 1789–1796. <https://doi.org/10.1016/j.eswa.2014.10.016>
- Tong, E. N. C., Mues, C., & Thomas, L. C. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1), 132–139. <https://doi.org/10.1016/j.ejor.2011.10.007>
- Tong, E. N., Mues, C., Brown, I., & Thomas, L. C. (2016). Exposure at default models with and without the credit conversion factor. *European Journal of Operational Research*, 252(3), 910–920.
- Tony Bellotti & Jonathan Crook. (2013). Forecasting and stress testing credit card default using dynamic models. *International Journal of Forecasting*, 29(4), 563–574.
- Torrent, N. L., Visani, G., & Bagli, E. (2020). PSD2 Explainable AI Model for Credit Scoring. *ArXiv:2011.10367 [Cs]*. <http://arxiv.org/abs/2011.10367>
- Tripathi, D., Edla, D. R., Bablani, A., Shukla, A. K., & Reddy, B. R. (2021). Experimental analysis of machine learning methods for credit score classification. *Progress in Artificial Intelligence*, 10(3), 217–243.
- Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, 34(4), 2639–2649.
- Tsai, C.-F., & Chen, M.-L. (2010). Credit rating by hybrid machine learning techniques. *Applied Soft Computing*, 10(2), 374–380. <https://doi.org/10.1016/j.asoc.2009.08.003>
- Tsymbol, A. (2004). The problem of concept drift: Definitions and related work. *Computer Science Department, Trinity College Dublin*.
- Turiel, J. D., & Aste, T. (2019). Peer-to-peer loan acceptance and default prediction with artificial intelligence. *Royal Society Open Science*, 7(6), 191649. <https://doi.org/10.1098/rsos.191649>

Turner, M. A., & Agarwal, A. (2008). Using non-traditional data for underwriting loans to thin-file borrowers: Evidence, tips and precautions. *Journal of Risk Management in Financial Institutions*, 1(2), 165–180.

Turner, M. A., Lee, A. S., Schnare, A. B., Varghese, R., & Walker, P. D. (2006). *Give Credit Where Credit is Due: Increasing Access to Affordable Mainstream Credit Using Alternative Data*. Information Policy Institute/Political and Economic Research Council. <http://dspace.cigilibrary.org/jspui/handle/123456789/5473>

Turner, M. A., Varghese, R., Walker, P. D., & Chaudhuri, Sukanya. (2012). *The Credit Impacts on Low-Income Americans from Reporting Moderately Late Utility Payments*. Information Policy Institute/Political and Economic Research Council. <http://dspace.cigilibrary.org/jspui/handle/123456789/5473>

Turner, M. A., Walker, P. D., & Dusek, Katrina. (2009). *New to Credit from Alternative Data*. Information Policy Institute/Political and Economic Research Council. http://www.perc.net/wp-content/uploads/2013/09/New_to_Credit_from_Alternative_Data_0.pdf

Uddin, M. F. (2019). Addressing Accuracy Paradox Using Enhanced Weighted Performance Metric in Machine Learning. *2019 Sixth HCT Information Technology Trends (ITT)*, 319–324. <https://doi.org/10.1109/ITT48889.2019.9075071>

Valverde-Albacete, F. J., & Peláez-Moreno, C. (2014). 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. *PLOS ONE*, 9(1), e84217. <https://doi.org/10.1371/journal.pone.0084217>

Verikas, A., Kalsyte, Z., Bacauskiene, M., & Gelzinis, A. (2010). Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey. *Soft Computing*, 14(9), 995–1010. <https://doi.org/10.1007/s00500-009-0490-5>

Vukovic, S., Delibasic, B., Uzelac, A., & Suknovic, M. (2012). A case-based reasoning model that uses preference theory functions for credit scoring. *Expert Systems with Applications*, 39(9), 8389–8395. <https://doi.org/10.1016/j.eswa.2012.01.181>

Wang, Q., Luo, Z., Huang, J., Feng, Y., & Liu, Z. (2017). A Novel Ensemble Method for Imbalanced Data Learning: Bagging of Extrapolation-SMOTE SVM. *Computational Intelligence and Neuroscience*, 2017, 1–11. <https://doi.org/10.1155/2017/1827016>

Wang, S., Minku, L. L., & 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. <https://doi.org/10.1109/TNNLS.2017.2771290>

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 359–372.

Wei, Y., Yildirim, P., Van den Bulte, C., & Dellarocas, C. (2015). Credit Scoring with Social Network Data. *Marketing Science*.
<http://pubsonline.informs.org/doi/abs/10.1287/mksc.2015.0949>

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, 27(11–12), 1131–1152. [https://doi.org/10.1016/S0305-0548\(99\)00149-5](https://doi.org/10.1016/S0305-0548(99)00149-5)

Whittaker, J., Whitehead, C., & Somers, M. (2006). A dynamic scorecard for monitoring baseline performance with application to tracking a mortgage portfolio. *Journal of the Operational Research Society*, 58(7), 911–921.
<https://doi.org/10.1057/palgrave.jors.2602226>

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1), 69–101.

World Bank Group. (2019). *Credit Reporting Knowledge Guide 2019*. World Bank.
<https://doi.org/10.1596/31806>

Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78, 225–241. <https://doi.org/10.1016/j.eswa.2017.02.017>

Xu, G., Duong, T. D., Li, Q., Liu, S., & Wang, X. (2020). Causality Learning: A New Perspective for Interpretable Machine Learning. *IEEE Intelligent Informatics Bulletin*, 20(1), 7.

Xu, L., & Amari, S. (2009). Combining Classifiers and Learning Mixture-of-Experts. In J. R. Rabunal, J. Dorado, & A. Pazos Sierra (Eds.), *Encyclopedia of artificial intelligence* (pp. 318–326). IGI Global.

Xu, R., Nettleton, D., & Nordman, D. J. (2016). Case-Specific Random Forests. *Journal of Computational and Graphical Statistics*, 25(1), 49–65.
<https://doi.org/10.1080/10618600.2014.983641>

Yang, B. H., & Tkachenko, M. (2012). Modeling exposure at default and loss given default: Empirical approaches and technical implementation. *The Journal of Credit Risk*, 8(2), 81.

Yao, P. (2009). *Hybrid Fuzzy SVM Model Using CART and MARS for Credit Scoring*. 392–395. <https://doi.org/10.1109/IHMSC.2009.221>

Yao, X., Crook, J., & Andreeva, G. (2015). Support vector regression for loss given default modelling. *European Journal of Operational Research*, 240(2), 528–538. <https://doi.org/10.1016/j.ejor.2014.06.043>

Yu, J.-M. (2018). *Sustainable co-training of mixture-of-experts for credit scoring of borrowers in social lending*. 20.

Yu, L., Wang, S., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications*, 34(2), 1434–1444. <https://doi.org/10.1016/j.eswa.2007.01.009>

Yu, L., Wang, S., Lai, K. K., & Zhou, L. (2008). *Bio-Inspired Credit Risk Analysis: Computational Intelligence with Support Vector Machines*. Springer Verlag.

Zafar, M. B., Valera, I., Rodriguez, M. G., & Gummadi, K. P. (2017). Fairness Constraints: Mechanisms for Fair Classification. *ArXiv:1507.05259 [Cs, Stat]*. <http://arxiv.org/abs/1507.05259>

Zhang, H., Berg, A. C., 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, 2126–2136. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1641014

Zhang, H., & Liu, Q. (2019). Online Learning Method for Drift and Imbalance Problem in Client Credit Assessment. *Symmetry*, 11(7), 890. <https://doi.org/10.3390/sym11070890>

Žliobaitė, I. (2009). *Learning under Concept Drift: An Overview* [Technical Report]. Faculty of Mathematics and Informatics, Vilnius University. <http://arxiv.org/abs/1010.4784>

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. https://doi.org/10.1007/978-3-319-26989-4_4