Expert Systems With Applications 104 (2018) 1–21

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An artificial intelligence system for predicting customer default in e-commerce

Leonardo Vanneschi[a,*], David Micha Horn[a], Mauro Castelli[a], Aleš Popovič[a,b]

[a] NOVA IMS, Universidade Nova de Lisboa, Lisboa 1070-312, Portugal
[b] Faculty of Economics, University of Ljubljana, Kardeljeva Ploščad 17, 1000, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 10 November 2017
Revised 19 February 2018
Accepted 13 March 2018
Available online 14 March 2018

Keywords:
Risk management
Credit scoring
Genetic programming
Machine learning
Optimization

ABSTRACT

The growing number of e-commerce orders is leading to increased risk management to prevent default in payment. Default in payment is the failure of a customer to settle a bill within 90 days upon receipt. Frequently, credit scoring (CS) is employed to identify customers' default probability. CS has been widely studied, and many computational methods have been proposed. The primary aim of this work is to develop a CS model to replace the pre-risk check of the e-commerce risk management system Risk Solution Services (RSS), which is currently one of the most used systems to estimate customers' default probability. The pre-risk check uses data from the order process and includes exclusion rules and a generic CS model. The new model is supposed to replace the whole pre-risk check and has to work both in isolation and in integration with the RSS main risk check. An application of genetic programming (GP) to CS is presented in this paper. The model was developed on a real-world dataset provided by a well-known German financial solutions company. The dataset contains order requests processed by RSS. The results show that GP outperforms the generic CS model of the pre-risk check in both classification accuracy and profit. GP achieved competitive classificatory accuracy with several state-of-the-art machine learning methods, such as logistic regression, support vector machines and boosted trees. Furthermore, the GP model can be used in combination with the RSS main risk check to create a model with even higher discriminatory power.

1. Introduction

E-commerce vendors in Germany have to deal with a peculiarity: commonly used payment types like credit cards and PayPal represent relatively low market shares, and the majority of orders are processed using open invoice instead. Using open invoice, a vendor bills customers for goods and services only after delivery of the product. Thus, the vendor grants customers a credit to the extent of the invoice. Usually, the vendor sends customers an invoice statement as soon as the products are delivered or provided. The invoice contains a detailed statement of the transaction. Because the customer receives a purchase before payment, it is called open, and the invoice is closed once the payment is received. Around 28% of customers in Germany choose open invoice as their payment type (Frigge, 2016), and around 68% of customers name open invoice as one of their favorite payment types (Fittkau & Maa Consulting, 2014; Wach, 2011). However, open invoice is prone to payment disruptions. Among the most common reasons, vendors find that customers simply forget to settle the bill or delay the payment on purpose. However, around 53% of vendors state that insolvency is one of the most common reasons for payment disruption (Weinfurner, Weisheit, Wittmann, Stahl, & Pur, 2011). The majority of the cases that conclude in default on payment in Germany are nowadays orders with open invoices, with more than 8% of all orders defaulting (Seidenschwarz, Weinfurtner, Stahl, & Wittmann, 2014). E-commerce vendors find themselves in a conflict: offering open invoice incentivizes many clients to confirm their purchases but, at the same time, increases the risk of default on payment rate. The former aspect has a positive effect on revenue, while the latter drives it down. Additionally, default on payment has a negative impact on the profit margin, due to costs arising through the provision of services and advance payments to third parties. In order to break through this vicious circle, vendors can fall back on a plethora of methods. Many tackle this conflict by implementing exclusion rules for customer groups they consider especially default-prone (for instance, customers who are unknown to the vendor or whose order values are conspicuously high). Another approach,

[*] Corresponding author.
E-mail addresses: lvanneschi@novaims.unl.pt (L. Vanneschi), dmhorn@novaims.unl.pt (D.M. Horn), mcastelli@novaims.unl.pt (M. Castelli), apopovic@novaims.unl.pt (A. Popovič).

https://doi.org/10.1016/j.eswa.2018.03.025
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

used by more than 30% of e-commerce vendors in Germany, is to fall back on external risk-management services (Weinfurner et al., 2011). Risk management applications are aimed at detecting customers with a high risk of defaulting. Those applications are frequently built using credit scoring (CS) models. CS analyzes historical data to isolate meaningful characteristics that are used to predict the probability of default (Mester, 1997). However, the probability of default is not an attribute of potential customers but merely a vendor's assessment of whether the potential customer is a risk worth taking. Over the years, CS has evolved from a subjective vendor's "gut" decision to a method based on statistically sound models (Thomas, Edelman, & Crook, 2002). Among the providers of risk management services in Germany is the risk management division of Arvato Financial Solutions (AFS), which provides a number of services, including identification of individuals, evaluation of credit-worthiness, and fraud recognition. The AFS databases consist of 21 million solvency observations totaling information from 7 million individuals in Germany, addresses and change in address information, and bank account information as well as phone numbers, email addresses, and device information. AFS's risk management service for e-commerce is called Risk Solution Services (RSS). RSS covers the entire order process and provides a number of services for every stage of the order process. The main service for evaluating customers' default probability is called risk check and is split into a pre-risk check and a main risk check. The main risk check is based on a credit agency score that uses country-specific solvency information on individuals. Hence, the main risk check is inoperable in countries without accessible solvency information. Contrarily, the pre-risk check was designed to always be operable and to ensure that the risk check returns an evaluation of the customers' default probability. For this purpose, the pre-risk check uses data transmitted by the customer during the order process. However, the pre-risk check in several industrial realities is nowadays based on a generic model, sometimes even without statistically sound backup (Lessmann, Baesens, Seow, & Thomas, 2015).

The objective of this work is to use genetic programming (GP) to build a CS model to replace the existing RSS pre-risk check. This is done in continuity with a precise recent research track, aimed at using technology to improve risk management (Lessmann et al., 2015). Inspired by Darwin's theory of evolution, GP (Koza, 1992a) is a computational intelligence (CI) method that employs evolutionary mechanisms such as inheritance, selection, crossover, and mutation to gradually evolve new solutions to a problem. In a CS environment, GP initializes a population of discriminant functions to classify customers into bad and good ones (hereafter called *bads* and *goods* for simplicity). This population is subsequently evolved to find the best possible discriminant function. The motivation for using a CI method to tackle the problem comes from Marques, Garcia, and Sanchez (2013), who discuss five major characteristics of CI systems that are especially appealing in CS: learning, adaption, flexibility, transparency, and discovery. Learning describes the ability to learn decisions and tasks from historical data. Adaption represents the capability to adapt to a changing environment, i.e., without being restricted to specific situations or economic conditions. The flexibility of CI systems allows for utilization even with incomplete or unreliable datasets. Furthermore, Marques and colleagues state that CI systems may be transparent, in the sense that resulting decisions may be visible and thus at least partially explainable in some cases. Lastly, discovery represents the ability to find previously unknown relationships. Inside the wide field of CI, our focus on GP follows the same motivations as in Ong, Huang, and Tzeng (2005), where it is argued that GP has a number of attractive characteristics for its application in CS. First, it is a non-parametric tool and is not restricted to specific situations or datasets, but can be used in a vast context. Second, it automatically determines the most fitting discriminant function. Last but not least, GP can automatically select the most important variables during the learning phase. Indeed, research has already shown the benefits of GP and its utility in CS (see Section 3 for a detailed discussion of the state of the art). However, CS is usually employed with data from the financial sector, while other sectors have rarely been considered so far. In this work - for the first time, to the best of our knowledge - we extend current research in CS by employing GP on a dataset that contains orders from e-commerce vendors.

This work is organized as follows. Section 2 contains a general introduction to the theoretical framework of CS. In Section 3, previous and related work is analyzed and discussed. Section 4 presents the RSS and the services it provides for every stage of the order process. In Section 5, we describe the dataset used in this work and provided by AFS. Section 6 presents the organization of our experimental study and a discussion of our experimental settings. In Section 7, we present and discuss the obtained experimental results. Finally, Section 8 concludes the paper and proposes ideas for future research. The paper is terminated by Appendix A, in which we briefly introduce GP for readers who are not familiar with this computational method and also suggest bibliographic material to deepen the readers' understanding of the subject.

2. Theoretical framework

CS is widely used by financial institutions to determine applicants' default probability and subsequently classify them into good applicants (the "goods", for simplicity) or bad applicants (the "bads") (Thomas et al., 2002). Consequentially, applicants may be rejected or accepted as customers based on that classification. Thus, CS represents a binary classification problem (Henley, 1995). The binary response variable represents a default in payment by the customer, or potential default in payment for order requests that have been declined. Article 286 (3) of the German Civil Code defines delay of payment as the non-settling of bills within a 30-day period (assuming no other payment target between the vendor and customer is contractually scheduled). Also, Article 178 (1b) of European Union Regulation 575/2013 considers a default as having occurred once an obligor is more than 90 days past due. Hence, a customer who did not settle his or her account within a period of three months upon receipt is considered to have defaulted in payment (Hand & Henley, 1997; Thomas, 2000). To evaluate the credit risk of loan applications, CS is targeted at isolating the effects of applicants' characteristics on their default probability. The default probability is mapped to a numerical expression usually called *score* that indicates the applicants' creditworthiness and enables the creditor to rank the applicants. The relationship between default probability and scores is depicted in Fig. 1. The mean scores for all of the applicants who perform well should be higher than the mean scores for all of the applicants who perform badly. The better the mean scores separate good from bad applicants, the higher the discriminatory power (Mester, 1997). Fig. 2 shows the distributions of bads and goods for four example models, with the scores on the x-axis and the number of observations on the y-axis. The bads are represented by the red line, whereas the goods are represented by the green line. The dashed line depicts the mean values for bads and goods. In model (a), bads and goods are non-overlapping; hence, the model has maximum discriminatory power. Model (b) shows slightly overlapping goods and bads. Thus, the discriminatory power is reduced compared to the previous case but still high. In model (c), goods and bads are heavily overlapping, which implies low discriminatory power. Finally, the mean scores of goods and bads in model (d) are identical; hence, the model incorporates no discriminatory power. The classification and consequent decision of whether to reject or accept an applicant are taken by comparing the applicant's CS with a predefined thresh-

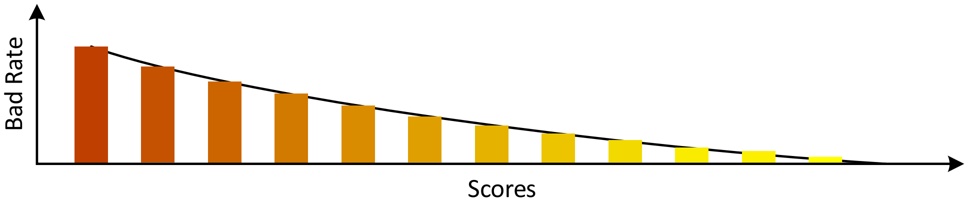

Fig. 1. Relationship between default probability and scores.

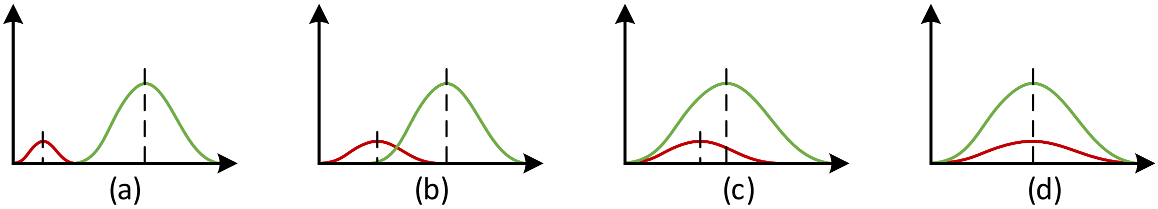

Fig. 2. Example distributions of goods and bads.

old. Thus, creditworthiness is not an attribute of the applicant but an assessment of whether the applicant represents a risk willing to be taken by the lender (Thomas et al., 2002). Scoring models are built upon historical data from applications, including application details and information about the true outcome of an accepted application, called performance. The benefits of a CS model for financial institutions are threefold (Mester, 1997). First, the application of a CS model reduces the time needed for the approval process. The possible time savings vary depending on how strictly the proposed cut-off threshold is followed. If the threshold is strictly followed, then credit applicants above or below the threshold will be automatically accepted or rejected, respectively. Otherwise, if the threshold is not followed strictly, the applications within a certain range around the threshold can be reevaluated. In either case, the efficiency of CS can improve greatly because applicants far from the threshold are automatically detected and categorized. Second, because applying and handling the application take less time, both parties save money. Lastly, both the creditor and applicant benefit from increased objectivity in the loan-approval process. Using a CS model ensures that the same criteria are applied to all applicants regardless of the creditor's personal feelings toward the applicant.

The objectives in CS for mail-order and e-commerce vendors differ from those of finance and banking. Rejected individuals are not denied from purchasing but usually just restricted to secure payment methods at a point of sale. Secure payment methods require the customer to pay the ordered goods before shipment or on delivery, e.g. advanced payment, credit card, pay on delivery, or PayPal. Denying open invoice and other insecure payment methods increases the abortion rate during the payment process. Individuals retained in the payment process are usually measured by the conversion rate, i.e., the ratio of individuals terminating the order process to individuals entering the payment process. Hence, CS in e-commerce is aimed at lowering default on payment by rejecting individuals with a high probability of default from purchasing goods using insecure payment methods while retaining a maximum conversion rate.

In Sackmann, Siegl, and Weber (2011), the risk of default in payment is divided into two dimensions. The first dimension is represented by individuals who are unable to settle their bills because of insolvency or illiquidity, and the second dimension is represented by individuals who are unwilling to settle their bills because they are engaging in fraud. Using those dimensions, four risk categories were defined:

1. Ability and willingness to settle bill
2. Inability to settle bill
3. Unwillingness to settle bill
4. Inability and unwillingness to settle bill

The effect of both dimensions for the vendor is the same: disruption or default in payment. The causes, however, differ and demand adjusted prevention approaches. Inability to pay due to insolvency or illiquidity is usually identified using classical CS models, as described above. Unwillingness to pay can be detected by identifying fraud patterns such as irregular order values and volumes. Fraud is also regularly committed using the account data of public institutions or charity organizations and by impersonating others, for which adequate prevention measures exist. Fraud prevention can also be integrated into CS models, which covers the fourth risk category (Sackmann et al., 2011). Additionally, those approaches may have a positive effect for individuals who enter their information wrongly by accident but are both able and willing to settle their bills. Denying them a purchase based on insecure payment methods may prompt them to review their input data and thus decrease wrong deliveries and running costs. The ability to combine the identification of those who are unable with those who are unwilling to settle their bills usually elevates CS to a very capable risk management tool in e-commerce.

3. Literature review

CS is currently a widely studied research field, and several important contributions have appeared. For a detailed survey of classification algorithms for CS, the reader is referred to Lessmann et al. (2015). While an attempt to exhaustively cover all existing contributions here is purely utopic, given the limited available space, we organize this section in the following way: in the first part, we present the history and evolution of the field, while in the last part, we focus on the most recent publications and those we consider more relevant for our work. Throughout this section, more attention is dedicated to discussing contributions that are in some ways related to our work, such as those describing the developments of computational methods for predictive modeling in CS, particularly evolutionary computation.

The history of CS goes back to mail-order companies (the predecessors of e-commerce vendors) in the 1930s. Credit decisions were made on the basis of subjective judgment by credit analysts. Thus, applicants may be rejected by one credit analyst but accepted by another. In an effort to diminish the respective inconsistencies, mail-order companies introduced a numeric scoring system. Only approximately 30 years later, such as in Myers and Forgy (1963), the first scoring systems to replace numerical systems were developed based on the pooled judgments of credit analysts. The decisive milestone in the history of CS was the Equal Credit Opportunity Act (ECOA) in the United States in 1974. Under the ECOA, race, color, national origin, religion, marital status, age, sex, and level of involvement in public assistance were prohibited from being used in CS. Furthermore, the ECOA classifies scoring systems into those that are “empirically derived” and those that are “statistically valid”. As a result, judgmental scoring systems were weakened, and scoring systems based on statistically valid models were generally accepted (Code, 1974; Mays, 2001; Thomas et al., 2002).

Over the years, a vast number of approaches to obtain a satisfactory CS model have been proposed. The classical approaches involve methods such as linear discriminant models (Reichert, Cho, & Wagner, 1983), logistic regression (Henley, 1995; Wiginton, 1980), k-nearest neighbors (Henley & Hand, 1996), and decision trees (Davis, Edelman, & Gammerman, 1992), but more sophisticated approaches such as artificial neural networks (Desai, Crook, & Overstreet, 1996; Malhotra & Malhotra, 2002; West, 2000) and GP (Abdou, 2009; Ong et al., 2005) have also been employed. One of the first published works using GP for CS (Ong et al., 2005) compared GP to several other machine learning methods (namely artificial neural networks, classification and regression trees, C4.5, rough sets, and logistic regression) using two credit datasets from Australia and Germany. Comparing the error rates, the authors showed that GP outperformed the other models in both datasets. The authors concluded that GP is a non-parametric tool that is not based on any assumptions concerning the dataset, which makes it suitable for potentially situation. Additionally, the authors stated that GP is more flexible and accurate than the compared techniques. These conclusions encouraged us to pursue the work, extending current research in CS by employing GP in the e-commerce domain for the first time. Later, Huang, Chen, and Wang (2007) investigated the CS accuracy of three hybrid support vector machines (SVMs). The SVM models were combined with the grid search approach to improve model parameters, the F-score for feature selection, and genetic algorithms (GAs) (Goldberg, 1989) (an evolutionary algorithm older than GP) to obtain both the optimal features and the parameters automatically, at the same time. The results of the hybrid SVM models were compared to other data-mining approaches based on artificial neural networks, GP and C4.5. The authors found that the best results were achieved by GP, followed by their hybrid SVM approach. They also pointed out that GP used fewer features than input variables, due to its automatic feature selection. Also, the results reported in Huang et al. (2007) were greatly encouraging for us to pursue our work. Also, Zhang, Huang, Chen, and Jiang (2007) compared GP, artificial neural networks, and SVMs in CS problems. Additionally, a model was constructed by combining the classification results of the other models, using majority voting. The classification accuracy was used as the evaluation criterion. The authors showed that the different models obtained good classification results, but their accuracies differed little. Furthermore, the combined model showed better overall accuracy. As we will show later, our work has led to similar conclusions, albeit using a problem domain and data that are very different from those in Zhang et al. (2007). In Abdou (2009), two GP models were compared with probit analysis and weight of evidence in the Egyptian public banking sector. The dataset was provided by Egyptian commercial public sector banks. Abdou used both the average correct classification rate and the estimated misclassification cost as the evaluation criteria. In his conclusion, Abdou stated that the preferred model depends on which evaluation criterion is used but that the results obtained by the different studied methods were, generally speaking, comparable to each other. A two-stage GP model was implemented in Huang, Tzeng, and Ong (2006) to harvest the advantages of function-based and induction-based methods, and used for CS prediction. In the first stage, GP was used to derive if-then-else rules; in the second stage, GP was used to construct a discriminative function from the reduced data. The two-stage split was implemented to reduce the model’s complexity and therefore ensured the comprehension of the decision rules. The results were compared to a number of data-mining techniques but showed no significant improvement of two-stage GP over plain GP. However, the authors pointed out that in addition to general GP advantages (like the above mentioned automatic feature selection and function derivation), two-stage GP also provided “intelligent” rules that can be independently used by a decision maker.

In the last few years, the number of contributions presenting computational methods for specialized predictive modeling in CS has grown significantly, testifying to a growing interest among researchers and practitioners in this area. For instance, Li, Tian, Li, Zhou, and Yang (2017) studied the reject inference problem. Reject inference is a technique used to infer the outcomes for rejected applicants and incorporate them into the scoring system, with the expectation that doing so will improve predictive accuracy. A new method involving machine learning to solve the reject inference problem was proposed, and a semi-supervised SVM model was found to improve the performance of scoring models compared to the industrial benchmark of logistic regression. In the same vein, in Yao, Crook, and Andreeva (2015), SVM regression was successfully applied to predict loss given default of corporate bonds. In Neto, Adeodato, and Salgado (2017), the preprocessing stage in CS was studied, i.e., the phase in which data are transformed for an effective later application of a data mining technique. The authors proposed a framework consisting of constructing new input variables that embed temporal knowledge. They demonstrated that the proposed preprocessing method was able to improve the discriminant power of some data-mining techniques. Also, they showed that the time of data transformation could be reduced using automatic code generation. Finally, they demonstrated that artificial intelligence and statistics modelers could effectively perform the data transformation without the help of database experts. In Luo, Kong, and Nie (2016), a regression spline-based discrete time survival model was applied with noteworthy results to CS risk management, stress testing, and credit asset evaluation. Ensembles of classifiers were applied to bankruptcy prediction and CS in Abelln and Mantas (2014). More specifically, a bagging scheme was applied to several decision tree models, with interesting results. In Kruppa, Schwarz, Arminger, and Ziegler (2013), a general framework for estimating individual consumer credit risks by means of machine learning methods was presented. Among others, random forests (RF), k-nearest neighbors (kNN), and bagged k-nearest neighbors (bNN) were successfully applied, as well as an optimized logistic regression.

The performance of a number of modelling approaches for the particular CS case of future mortgage default status was investigated in Fitzpatrick and Mues (2016). More specifically, boosted regression trees, random forests, and penalized linear and semi-parametric logistic regression were applied to four portfolios of over 300,000 Irish owner-occupier mortgages. The main findings were that the selected approaches have varying degrees of predictive power and that boosted regression trees were able to significantly outperform logistic regression. The conclusion was that boosted regression trees could be a useful addition to the current toolkit for mortgage credit risk assessment by banks and regulators.

Mixture cure models were introduced to CS in Tong, Mues, and Thomas (2012). A mixture cure model was used to predict the (time to) default on a UK personal loan portfolio and was compared to the Cox proportional hazards method and to standard logistic regression. The discrimination performance for all three ap-

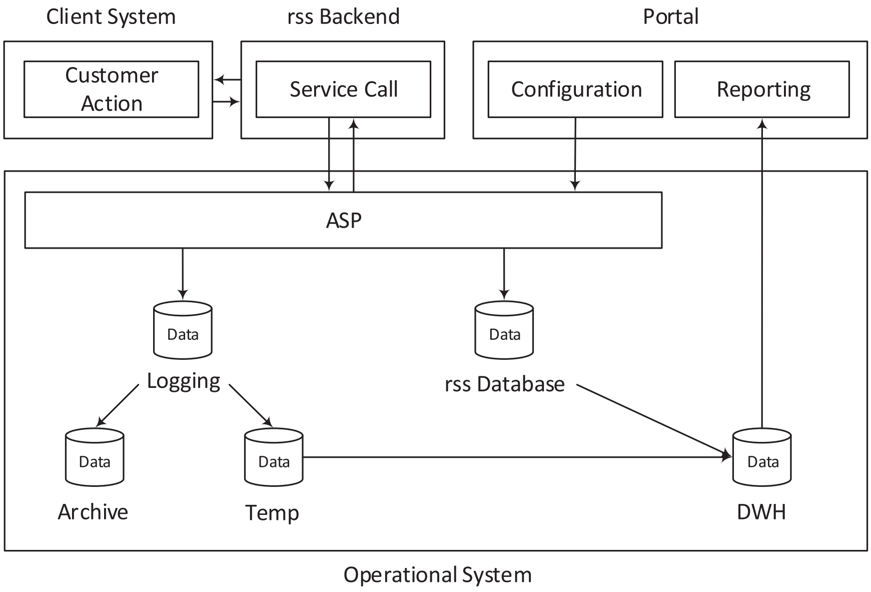

Fig. 3. A graphical representation of the RSS system.

proaches was found to be high and competitive. The calibration performance for the survival approaches was found to be superior to that of logistic regression in several case studies. Furthermore, the mixture cure model's ability to distinguish between two subpopulations was shown to offer additional insights by estimating the parameters that determine susceptibility to default, in addition to parameters that influence a borrower's time to default. In the same vein, Alves and Dias (2015) introduced a general framework of survival mixture models that addresses the heterogeneity of the credit risk among a financial institution's clients. This framework was able to identify clusters or groups of clients with different risk patterns. Among the studied methods, the time between the first delayed payment and default was best modeled by a three-segment log-normal mixture distribution and a multinomial logit link function. The model was able to predict the most likely risk segment for each new client. In Verbraken, Bravo, Weber, and Baesens (2014), a profit-based classification performance measure to credit risk modeling was presented. This performance measure is based on the expected maximum profit (EMP) and is used to find a trade-off between the expected losses driven by the exposure of the loan and the loss given default and the operational income given by the loan. The authors state that one of the major advantages of using the proposed measure is that it permits the optimal cutoff value to be calculated, which is necessary for model implementation. The presented results show that the proposed profit-based classification measure allowed some classifiers to outperform the alternative approaches in terms of both accuracy and monetary value in the test set. A different framework for CS modelling was presented in Han Ju and Young Sohn (2014). This framework consisted of ways to construct new technology CS model by comparing alternative scenarios for various relationships between existing and new attributes, based on explanatory factor analysis, analysis of variance, and logistic regression.

In this paper, we follow the suggestions in Lahsasna, Ainon, and Wah (2010), which recommended more cooperation between academic researchers and financial institutions, as well as the study of new informative data, for CS. Indeed, our work resulted from active collaboration between industry and academia, and a real-world dataset was used for the first time. The dataset contains order requests from e-commerce vendors and was provided by AFS.

4. Risk solution service

Risk Solution Service (RSS) is a risk management service that aims to cover the whole order process of e-commerce retailers' customers. Its objectives are threefold. First, increase conversion rate and customer retention in the e-shop by improving differentiation and managing of payment methods. Second, enhance cost control by providing innovative pricing models and configurable standard solutions in different service levels. Third, improve discriminatory power by combining current and historical customer information.

The functioning of the RSS system is shown in Fig. 3. Customer actions are registered by the client system and passed on to the RSS backend. Depending on the customer action, service calls are triggered as depicted in Fig. 4. The data content of the service calls is passed on to the ASP platform, which carries out the scoring of the customer. The ASP platform saves the request data in a logging database from where it is passed on to an archive and a reporting database. From the latter, the data is passed on to the data warehouse and merged with additional information from an extra RSS database. The reporting system in the portal obtains the data from the data warehouse. Additionally, the client-specific configuration is retained on the portal for easy access by the client. On every data transfer, the data is reduced and partly transformed. Thus, the data in the data warehouse does not entirely match the data used in the service calls.

RSS consists of a number of different risk management services, whose usage depends on the current customers' order process stage, as mapped in Fig. 4. Before the order process starts, upon registration of the clients' customer, RSS offers an address check. The account check verifies the validity of the entered address data by validation of correct postal syntax, verification and, if necessary, correction of mail address, and verification of deliverability by checking against a list of known addresses. Once a customer places an order with the client, a risk check is performed. The risk check returns a risk assessment to the client. Based on the risk assessment, the client offers his customer selected payment methods from which the customer may choose whichever he prefers. In case the customer chooses direct debit, an account check is performed in order to control the customer's submitted account data. The account check verifies the validity of the en-

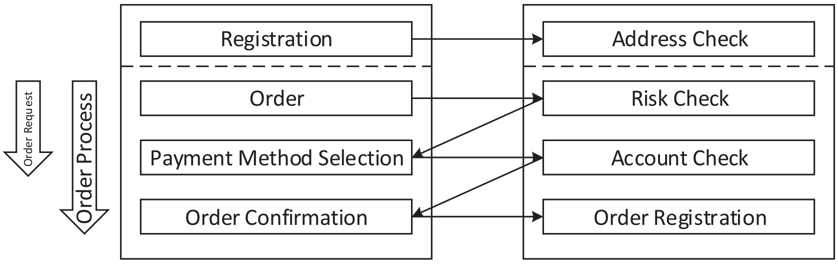

Fig. 4. RSS risk management services.

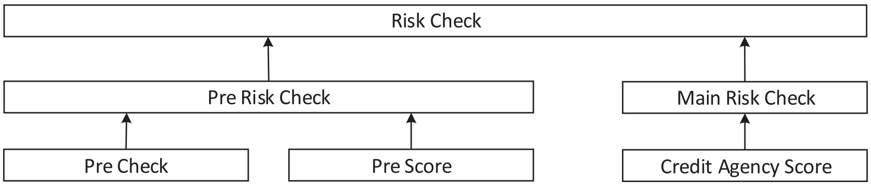

Fig. 5. Risk check.

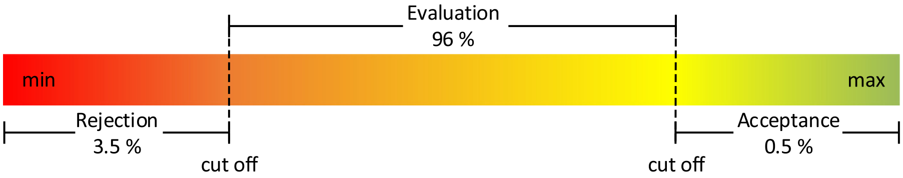

Fig. 6. Pre-score behavior in score range.

tered account data by validating correct syntax, comparing with whitelists of existing bank identification numbers and comparing with blacklists of publicly visible bank accounts, e.g., public institutions or charity organizations. In case the customer is accepted, an order confirmation is transmitted and the order gets registered in the RSS system.

RSS is designed to work not only with the scoring system provided by AFS but also with the scoring systems of other providers. Therefore, in order to also ensure operationality in the absence of the AFS credit agency score, the risk check was split into two modules that can work in combination or on their own. The risk check consists of the pre-risk check and the main risk check, as depicted in Fig. 5. The former is a combination of a pre-check and a pre-score, and the latter calls different credit agency scoring systems. Added together, they form the RSS score. The RSS pre-risk check was developed in 2011, and a vast number of ad hoc updates were deployed over the years. However, neither the original design nor any update was based on a statistically sound model. Instead, a generic scorecard was developed, which is generally done whenever data is unavailable.

The pre-risk check consists of a pre-check and a pre-score. The former is a combination of exclusion rules that lead to the rejection of a request if triggered or otherwise to a pass on to the pre-score, which assigns a score according to the request data. The latter is a value centered at 0, which works as an “on top” addition or deduction to the credit agency score. The pre-check matches requests with a blacklist, compares the request basket value to a predetermined limit and employs a spelling and syntax review of the request input data. Around 30% of the requests are usually rejected by the pre-check and thus are not assigned a pre-score. The pre-score is based on a score card that distributes score points for a number of distinct features. Within a certain score range, the assigned score triggers one of three possible actions as depicted in Fig. 6. The customer is rejected below a certain threshold, and the customer is accepted above a certain threshold. Between those thresholds, the customer is passed on to the main risk check for further evaluation.

Taking into account that the pre-risk check was designed to be operated both with and without the main risk check, the design of the pre-risk score seems to incorporate a critical flaw. The requests that are passed on for further evaluation require the main risk check; otherwise, those requests have to either be accepted or rejected using other criteria. So far, around 96% of the requests evaluated by the pre-score were passed on to the credit agency score, around 0.5% of the requests were accepted and around 3.5% of the requests were rejected. Therefore, the overwhelming majority of requests requires additional evaluation.

Fig. 7 shows the deficit in discriminatory power of the pre-score. The score values are not spread apart over the score ranges, but heavily clumped together in few score groups that contain a majority of observations. Additionally, both good and bad requests are centered around a similar score range, with the number of good requests being continuously higher than the number of bad requests. However, the default rate shows a difference over the score ranges: the default rate is higher in the rejected range than in the range up for further evaluation, and the default rate in the latter is higher than in the accepted range. Both the rejection and the acceptance area have a negative trend, but the area up for further evaluation has a positive trend. Accordingly, the default rate increases as score values increase between the threshold score for rejection and acceptance, whereas increasing score values are supposed to map the decreasing probability of default. As a result, the pre-score lowers the discriminatory power of the credit agency score it is merged with, assuming the latter works properly.

The RSS pre-risk check provides an important part of the RSS system. It is designed to work both in combination with a credit agency score within the main risk check and on its own. However, the current implementation reduces the discriminatory power of

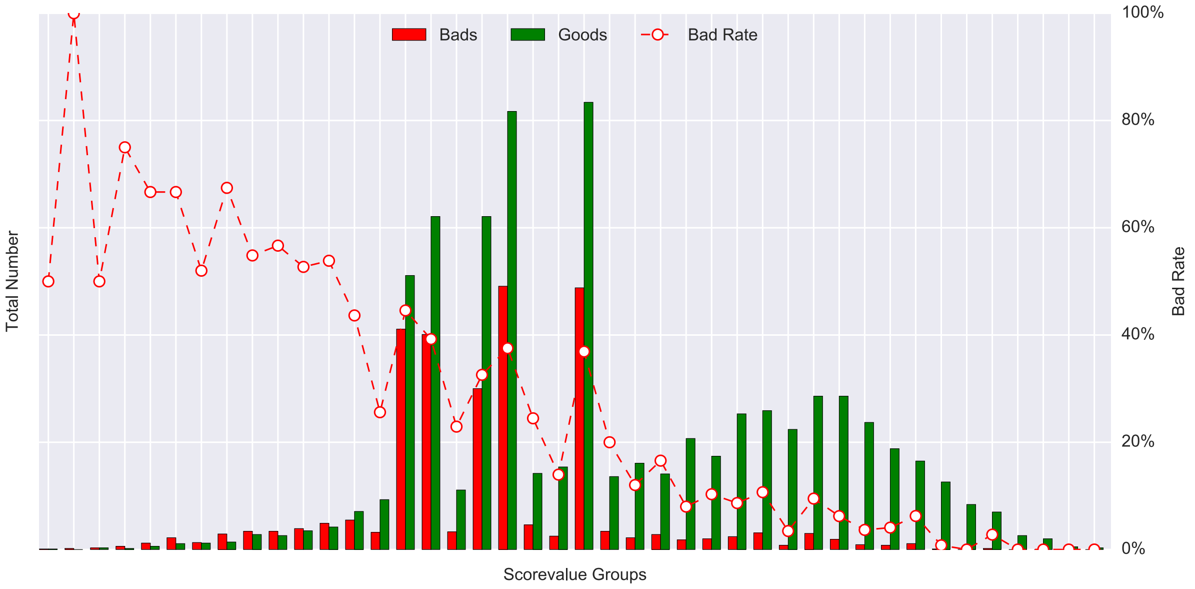

Fig. 7. Pre-score distribution of goods and bads.

the main risk check and offers little benefit on its own. The objective of this work is to revise the pre-risk check and to incorporate both pre-check and pre-score into a single score with a higher discriminative power than the existing pre-score. Additionally, the score needs to be operational on its own for usage in an international setting and in combination with a credit agency score.

5. Dataset

The dataset used in this work consists of order requests processed by RSS between 10-01-2014 and 12-31-2015, and it is provided by the AFS company. It contains 56,669 order requests, among which 15,535 ($\approx$ 27%) are labeled as "bad", while the remaining 41,134 are labeled as "good". These order requests are subject to a stratified random split into a training set with 31,669 ($\approx$ 56%) observations, a test set with 10,000 ($\approx$ 18%) observations and a validation set with 15,000 ($\approx$ 26%) observations (the validation set is needed because it is used in the calibration phase of our method, which will be described in Section 7.2). Data include personal information about the individuals that are processed. To account for the sensitivity of the data, i.e., to protect data confidentiality and to concur with data protection regulation, alterations were conducted. Therefore, the variables were renamed to IN0 – IN18. What follows is a brief description of those variables:

- IN0: binary variable with information about whether a customer is known to the client.
- IN1: binary variable with information about whether shipping address and billing address match.
- IN2: elapsed time in days since the customer was registered by the system for the first time.
- IN3: validation of the billing address, i.e., whether the address exists and a customer is known to live at the stated address.
- IN4; IN5; IN10; IN11; IN13; IN14; IN15; IN18: information about the customer's order history.
- IN6; IN7; IN8: information about the dunning history of a customer.
- IN9: discretized order value in relation to the clients' average order value.
- IN12: variable designed as means of fraud prevention. A fraudulent customer may employ a couple of orders with little order value to be known by the system as a good customer and subsequently employ expensive orders using insecure payment methods. Therefore, the order value is compared to the average of past orders by the customer.
- IN16: clock hour of order time.
- IN17: age of the customer in years.

In order to efficiently obtain the discriminant function, the continuous variables need to be discretized (Ong et al., 2005). The discretization is done using the weight of evidence (WoE) measure to assess the predictive power of each attribute. The WoE measures the difference between the distribution of goods and bads and therefore represents an attributes power in separating good and bad observations (Siddiqi, 2006). The WoE is frequently used in CS to convert a continuous variable into a discrete variable, and it is currently in use in the AFS company. The WoE is calculated as follows:

$$\mathrm{WoE} = \left[\ln\left(\frac{\text{Distribution Good}}{\text{Distribution Bad}}\right)\right] \times 100 \tag{1}$$

The idea is to partition the possible values of a continuous variable into bins. A bin with a WoE around zero has the same probability of observing a default in payment as the sample average. Contrarily, if a bin has a WoE above or below zero, the probability of observing a default in payment is above or below sample average, respectively. However, more important than the absolute WoE is the difference between bins. The variables' predictive power relates to the WoE difference between bins. With monotonically increasing WoE, the probability of default is always higher in a bin that contains larger values, and the opposite holds with monotonically decreasing bins (Siddiqi, 2006). The interval boundaries of the bins are selected in such a way that the bins are monotonically increasing or decreasing; the differences in WoE over the bins are roughly equal; the bins are as evenly sized as possible; and a minimal number of observations is maintained in each bin. Missing values are not submitted to the bin selection phase, but treated as a separate bin exhibiting a value of -1. Missing data is treated this way to account for its predictive value. Table 1 shows the discretization of continuous variables, together with the value ranges for the respective bins. Fig. 8 shows the discretized variables with their WoE values for every bin. The variables are ordered decreasing by their information value (IV). The IV is a measure of the discriminative power of the variable. The lower the IV of a variable,

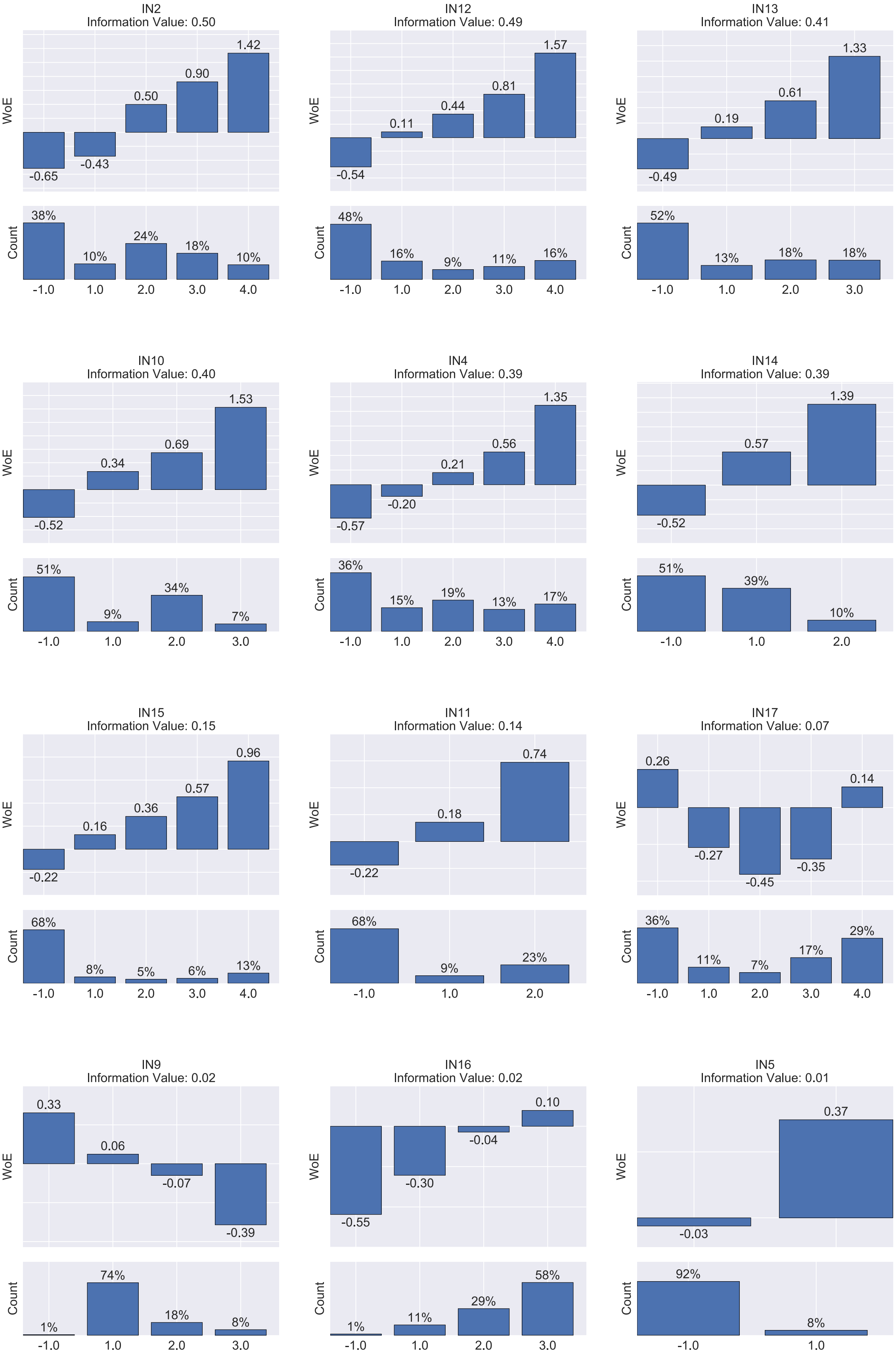

Fig. 8. Discretization plot.

Table 1
Discretization of continuous variables.

Variable	−1	1	2	3	4
IN2	∅	$[-\infty, 90)$	$[90, 380)$	$[380, 600)$	$[600, \infty)$
IN9	∅	$[-\infty, 150)$	$[150, 300)$	$[300, \infty)$	
IN17	∅	$[-\infty, 25)$	$[25, 30)$	$[30, 40)$	$[40, \infty)$
IN14	∅	$[-\infty, 18)$	$[18, \infty)$		
IN15	∅	$(\infty, 7)$	$[7, 4)$	$[4, 2)$	$[2, -\infty)$
IN13	∅	$[-\infty, 1)$	$[1, 4)$	$[4, \infty)$	
IN4	∅	$[-\infty, 1)$	$[1, 4)$	$[4, 10)$	$[10, \infty)$
IN5	∅	$[-\infty, \infty)$			
IN10	∅	$[-\infty, 5)$	$[5, 20)$	$[20, \infty)$	
IN11	∅	$(\infty, 6)$	$[6, -\infty)$		
IN12	∅	$(\infty, 50)$	$[50, 25)$	$[25, 10)$	$[10, -\infty)$
IN16	∅	$[-\infty, 9)$	$[9, 14)$	$[14, \infty)$	

Table 2
Input variables contained in the dataset used in this work. The leftmost column reports the name of the variable, while the remaining three columns report the overall mean of the variable, calculated on the whole dataset; the mean of the variable calculated only on the instances labeled as "good"; and the mean of the variable calculated only on the instances labeled as "bad", respectively.

Variable	Overall mean	Mean in "goods"	Mean in "bads"
IN0	0.47	0.39	0.69
IN1	0.05	0.05	0.06
IN2	1.13	1.45	0.26
IN3	−0.48	−0.56	−0.26
IN4	1.25	1.54	0.45
IN5	−0.83	−0.82	−0.88
IN6	0.26	0.23	0.35
IN7	−0.41	−0.44	−0.33
IN8	1.08	1.4	0.24
IN9	1.32	1.3	1.39
IN10	0.46	0.71	−0.2
IN11	−0.12	0	−0.45
IN12	0.84	1.18	−0.07
IN13	0.5	0.76	−0.21
IN14	0.09	0.27	−0.4
IN15	0.21	0.39	−0.28
IN16	2.43	2.46	2.34
IN17	1.57	1.51	1.72
IN18	−0.5	−0.43	−0.71

Table 3
GP parameter settings.

Parameter	Value
Population size	300
Initialization	Ramped half-and-half
Fitness function	Area under the ROC Curve
Function set	$\{+, -, \times, \div, \leq, \geq, =, \neq, if - then - else\}$
Terminal set	$\{Attributes, \frac{p}{10} \mid p \in \mathbb{N} \wedge p < 10 \cup x \in \mathbb{Z} \wedge -1 \leq x \leq 4\}$
Maximum number of generations	500
Selection	Tournament
Crossover rate	0.9
Mutation rate	0.1

the lower its predictive power. As a rule of thumb, variables with an IV of less than 0.05 do not add a meaningful predictive power to the model (Hand & Henley, 1997; Henley, 1995). The IV is calculated as follows:

$$\mathrm{IV} = \sum\nolimits_i (\text{Distribution Goods}_i - \text{Distribution Bads}_i) \times \ln\left(\frac{\text{Distribution Goods}_i}{\text{Distribution Bads}_i}\right) \tag{2}$$

Over the 12 discretized variables, three have an IV smaller than 0.05. Those variables are 'IN9', 'IN16' and 'IN5'. Nonetheless, they are not removed from the dataset, because while they seem insignificant by themselves, they might become important in interaction with others. The final dataset is described in Table 2.

6. Experimental organization and settings

When GP is employed to solve complex problems, like the one tackled in this paper, the use of an appropriate fitness function is often a crucial step. In this work, after considering several other possible measures, we have decided to use the area under the receiver operating characteristic (ROC) curve (ROC-AUC). ROC-AUC is the single-scalar representation of the ROC curve (Abdou & Pointon, 2011). The ROC curve is used when a classifier returns a numeric value that has to be interpreted as a class label using thresholds. A typical example for binary classification into two classes C_1 and C_2 is to interpret all positive outputs of the classifier system as categorizations of the observation into class C_1 and all other outputs as categorizations into class C_2. In this case, the threshold is set to zero. The ROC curve is a two-dimensional graph that quantifies the performance of a classifier in reference to all possible thresholds (Yang, Wang, Bai, & Zhang, 2004). For this purpose, the relative frequency of every unique score value specified by the True Positive Rate (TPR) (i.e., the proportion of actual positives predicted as positive) and the False Positive Rate (FPR) (i.e., the proportion of actual negatives predicted as negative) is plotted on a bidimensional Cartesian plane (Baesens et al., 2003). Therefore, the ROC graph shows relative trade-offs between benefits and costs. In order to compare different models, the ROC curve can be reduced to its single-scalar representation by calculating the ROC-AUC using Equation (3) (where T is a threshold parameter).

$$\text{Area under the ROC curve (ROC-AUC)} = \int_{-\infty}^{\infty} \mathrm{TPR(T)FPR'(T)\,dT} \tag{3}$$

A classifier with a ROC-AUC below 0.5 performs worse than a random classifier. However, since the ROC space is symmetrical around the diagonal line of the random classifier, negating a classifier below 0.5 produces its counterpart above 0.5. Overall, the ROC-AUC value ranges between 0.5 for a random classifier and 1 for a perfect classifier.

Motivations for using the ROC-AUC as a fitness for our GP systems are several, but the main one is that, in the application studied here, both TPR and FPR are important and have to be taken into account when evaluating the quality of a solution. Furthermore, ROC-AUC is suitable to be used as fitness because, as mentioned, it captures some of the characteristics of the ROC curve with just one scalar value. Also, the size of the model is an important factor for computational complexity and interpretability, and the size of the individuals evolved in the population may have an important impact on the efficiency of the system in terms of execution time and computational resources used. For this reason, we also decided to implement a simple mechanism of parsimony pressure, exactly as in Poli, Langdon, and McPhee (2008a): when two individuals have the same value of the ROC-AUC, preference is given to the smallest one (in terms of number of nodes in the tree) in the selection phase.

Beside fitness, GP also demands the setting of several other parameters. Table 3 reports the values used in our experiments. These values have been obtained as the result of a parameter tuning phase.

As mentioned above, ROC-AUC is the main measure of discriminatory power, and it is used as the GP fitness function. However, the ROC curve is insensitive to class distributions and does not account for the imbalance in datasets. On the other hand, being able to deal with imbalanced datasets is important for a predictive model. Therefore, when the experimental results will be presented, later in this paper, the ROC-AUC will not be the only reported measure of performance. In addition, we will report the results relative to the area under the Precision-Recall (PR) diagram, called PR-AUC. The importance of PR-AUC stands in the fact that, in CS, more customers pay their bills than those who do not pay them. Therefore, datasets are naturally unbalanced because of the abundance of goods that can be observed, compared to the bads. In these cases, moving the threshold to gain few more true positives may have a large impact on the number of false positives. As a result, FPR increases slightly, but TPR increases a lot more. Such a classifier receives its predictive power by rejecting a lot of goods in order to gain few bads. The larger the imbalance toward goods, the larger the concession in terms of falsely rejected goods.

A good classifier is able to obtain both high values for precision and high values for recall. However, precision and recall incorporate an intuitive tradeoff. Assuming that a higher score translates to a lower probability of default and vice versa, the trade-off can be generalized as follows. Lowering the threshold means to classify bads with a higher probability of default, which increases the precision. Additionally, the number of classified bads decreases, which decreases the recall as well. Subsequently, high precision and low recall translate to a high probability of default (low score). In such a model, those observations that are classified as bads are, so to say, “largely bads”, but many bads are misclassified as goods. Contrarily, low precision and high recall translate to a low probability of default (high score). In such a model, those observations that are classified as goods are “largely goods”, but almost all bads are classified as goods. Simplified, the more bads a model classifies as bads, the more goods are classified as bads as well. In a PR diagram, one classifier A is considered better than another classifier B if, for every recall value, the precision of A is higher than the precision of B.

Optimizing the ROC-AUC, in principle, does not optimize the PR-AUC too, despite the close relationship between the ROC space and the PR space. Hence, a classifier with excellent ROC-AUC may have poor PR-AUC. It is worth stressing again that only the ROC-AUC is used by our GP system to measure fitness. The value of the PR-AUC of the obtained model is also reported, but it was not used by the algorithm. The PR-AUC values are only reported in order to give a vision of the ability of the obtained model to deal with unbalanced datasets.

In our experiments, we performed 50 independent GP runs. At each of these runs, a different partition of the data into training, test, and validation was randomly generated and used. In the next section, when we will refer to the results obtained on the test set, we will refer to the performance on the test set of the best solution that GP (as well as the other methods used for comparison) was able to obtain on the training set. The outputs of the model learned by GP are scores, and so they do not represent probabilities of default. On the other hand, in CS, customers are typically sorted against the probability of default and accepted or rejected according to their relative position in respect to a threshold. For this reason, having probabilities is very important in CS. Calibrated scores allow us to interpret the score values as probabilities. This is a very important step because, with uncalibrated scores, it may happen, for instance, that a better score value inhibits a higher default in payment. Also, with uncalibrated scores, it may happen that a little difference between score values corresponds to a big difference in the probability of default in some cases but to little difference in the probability of default in others. In this work, the results are calibrated using isotonic regression. In order to validate the performance of isotonic regression, stratified k-fold cross-validation is applied to the calibrated training set, which was previously used as the validation set. The folds are stratified in order to retain the bad rate over all folds. Following Kirschen, O’Higgins, and Lee (2000), k is equal to 10 to keep bias and variance low while maintaining a reasonable number of observations in every fold. In the process of cross-validation, the dataset is divided into 10 non-overlapping subsets. In every iteration, nine subsets are allocated to train the model, and the remaining subset is used to test the model. The process is repeated 10 times, and thus every observation is in the test set once and nine times in the training set. The estimates over all iterations are averaged, thus generating the final output (Giovanni & Elder, 2010). The test set is never subject to any learning, but only to model application.

In the next section, the calibrated results are used to analyze the discriminatory power of the GP model on its own, and in combination with the already existing and currently used credit agency score of the AFS company, named ABIC. Additionally, models generated by other machine learning methods, like logistic regression, support vector machines and boosted trees are used for comparison against GP. All models are developed using the Python programming language, version 2.7, with the modules DEAP for GP and scikit-learn for Logistic Regression, Support Vector Machines and Boosted Trees (Fortin, De Rainville, Gardner, Parizeau, & Gagné, 2012; Pedregosa et al., 2011a). The overall work flow of our experimental study is represented in Fig. 9.

7. Experimental results

The presentation of the experimental results is organized as follows: in Section 7.1, we present the results obtained by GP in the CS problem described so far, and we dedicate particular attention to a discussion and an interpretation of the best model evolved by GP. In Section 7.2, we discuss the results we have obtained in the calibration phase. In Section 7.3, we compare GP and other machine learning methods. Finally, in Section 7.4, we discuss the results we obtained when GP was first compared to, and then used in collaboration with, the credit agency score.

7.1. Applying genetic programming. an analysis of the evolved predictive model

Fig. 10 shows the GP median best fitness against generations over 50 independent runs on the training set. The figure shows the typical trend of the evolution of fitness in GP: a rapid improvement in the first phase of the evolution, followed by a slower improvement in the later phase of the evolution (typically respectively corresponding to the phases of exploration and exploitation of the evolutionary process (Poli et al., 2008a)). More specifically, we can observe that the fitness increases heavily for the first ≈200 generations, while for the later ≈300 generations, fitness increases more slowly. Fig. 11 reports the best model evolved by GP in all the performed simulations. The model has a bushy shape and a size equal to 220 tree nodes, with a tree depth equal to the maximum allowed depth limit (i.e., the tree depth is equal to 17). Bushy models are frequently observed in GP using ramped half-and-half as the initialization method (Poli et al., 2008a). The two subtrees of the root are very different in size, with 193 and 27 nodes, respectively. Examining the tree, we can observe that pruning can be employed at a number of nodes to reduce the size of the model. Pruning the model yields a size of 154 nodes and a maximal tree depth equal to 16. Most noticeable, the if-then-else nodes yield no utility, but additional size, because every if-then-else node is connected to a boolean True/False decision terminal. Those nodes can be immediately replaced by their offspring node; for instance, an expression like $if-then-else(False, \mathrm{IN12}, \mathrm{IN16})$ can be replaced by IN16. Additionally, some subtrees consist of operations between constant terminals, which can be reduced to a single constant terminal node, e.g., $\times(0.7, 0.3)$ can be replaced by 0.21. The resulting constant terminal nodes 0.12, 0.21, and 5 depict values that are not available in the terminal set, from which only the values 0.3, 0.5, 0.7, and 2.0 are used.

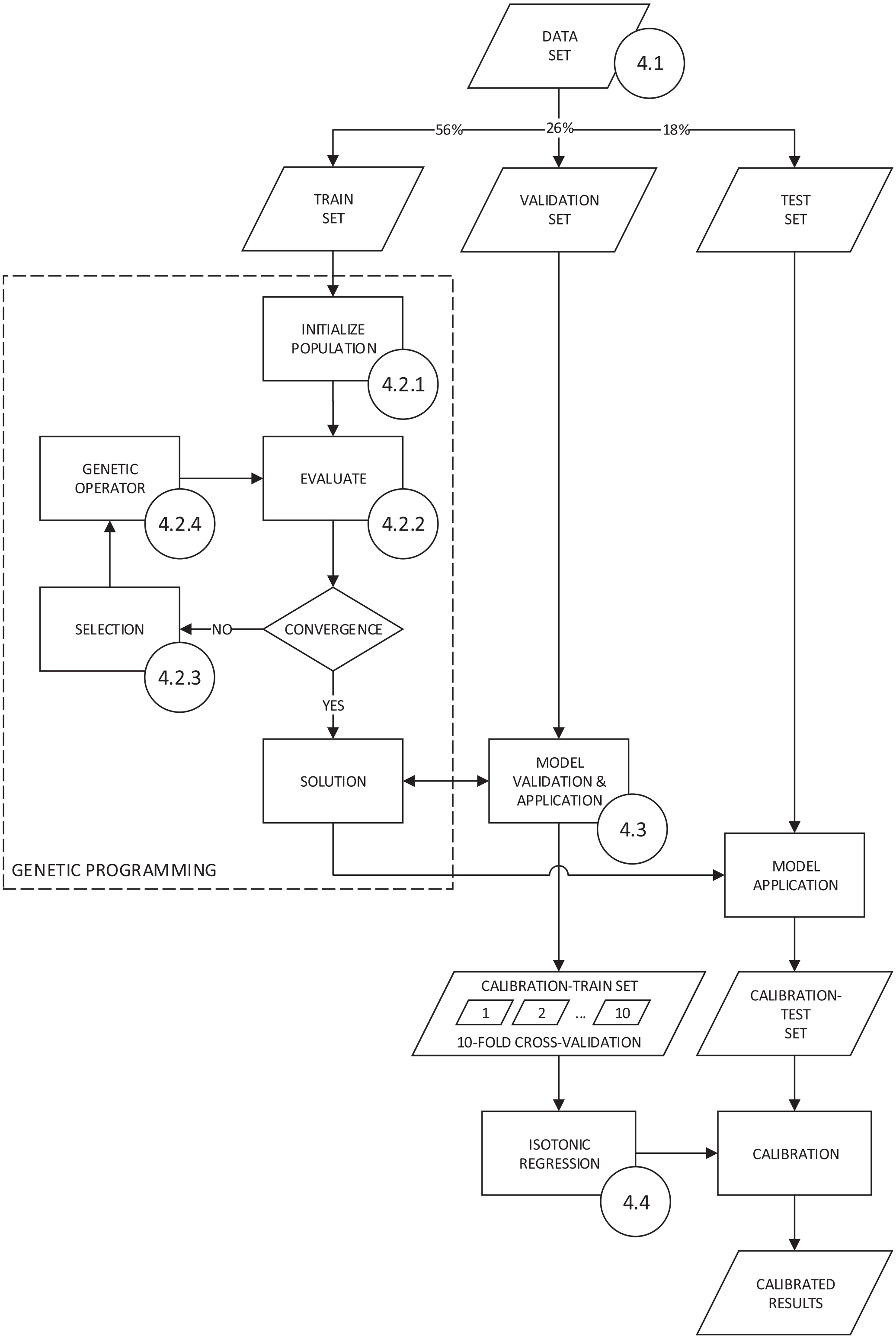

Fig. 9. Overall work flow. In this figure, the parallelograms represent datasets, while the rectangles represent the different processes that are performed during the execution of our system.

Because GP has the ability to automatically select features while learning the model, it is customary to conclude (see, for instance, Iba, Garis, & Sato, 1994) that the number of occurrences of a single variable in the final model gives information about the predictive power of that variable. From the 19 variables in the terminal set, six are not used in the model or are used in combination with the if-then-else nodes on the unused portion of the model. Those variables are IN1, which contains information about whether shipping address and billing address match, as well as variables IN5, IN11, IN14, IN15, and IN18, which contain information about the order history of a customer. The remaining order history variables, IN10 and IN13, are incorporated into the model but only oc-

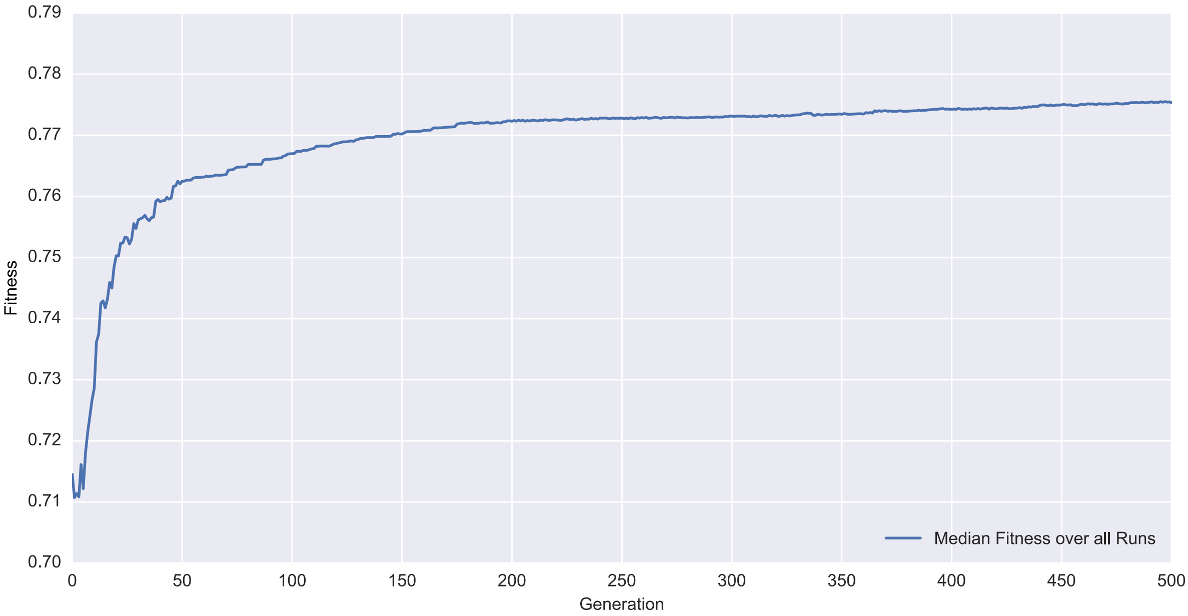

Fig. 10. Median fitness over all runs.

Table 4
Occurrences of variables.

Variable	Number of occurrence
IN7	13
IN2	10
IN6	9
IN12; IN16	5
IN3; IN4; IN17	3
IN0; IN8	2
IN9; IN10; IN13	1
IN1; IN5; IN11; IN14; IN15; IN18	0

cur one time in the model. Therefore, we can conclude that order history provides poor discriminatory power. Among the most often used variables are, instead, IN7 and IN6, with thirteen and nine occurrences, respectively. These variables, together with IN8, which occurs twice, contain information about the dunning history of a customer. A logical conclusion is that customers with prior payment difficulties are more likely to default than those without. With 10 occurrences in the final model, another heavily used variable is IN2, which contains information about the number of elapsed days since a known customer was registered by the system for the first time. This puts emphasis on continuous business connections between client and customers. Long-time customers are less likely to default on payment than first-time customers. Finally, the fraud-prevention variables IN12 and IN16, which contain the order time, occur five times. The remaining variables are used between one and three times. Table 4 shows the variables with their respective occurrences in decreasing order.

7.2. Calibration

The output of the model generated by GP is a score, and it can be used, for instance, to create a ranking of customers. However, the model's output does not represent the probability of default. Calibration is the phase that allows us to transform the outputs so they can be interpreted as probabilities. Probabilities are needed for two reasons. First, the GP outputs are supposed to be used as an on-top addition to or deduction from the credit agency score, which represents a probability; second, the outputs are eventually not going to be used in isolation but in combination with misclassification costs (Zadrozny & Elkan, 2001).

In order to calibrate the GP outputs, a non-parametric type of regression called isotonic regression (IR) is used. Among the model's benefits is the lack of assumptions about the target function's form. The IR problem is defined by finding a function that minimizes the mean-squared error (Niculescu-Mizil & Caruana, 2005; Zadrozny & Elkan, 2002):

$$\min \sum_i (y_i - f(x_i))^2 \tag{4}$$

IR uses pair-adjacent violators (PAVs) to find the best fitting stepwise-constant isotonic function, which works as follows. For every observation x_i, the value of the function to be learned $f(x_i)$ must be bigger or equal to $f(x_{i-1})$. Otherwise, the resulting function, f, is not isotonic, and the observations x_i and x_{i-1} are called pair-adjacent violators, hence the name of the algorithm. In order to restore the isotonic assumption, $f(x_i)$ and $f(x_{i-1})$ are replaced by their averages. This process is repeated with all observations until no PAVs remain (Ayer et al., 1955). In the setting of CS, PAV works as follows. First, the observations are ordered according to their score values. Then the $f(x_i)$ is set to 0 if x_i belongs to the goods, and $f(x_i)$ is set to 1 if x_i belongs to the bads. If the score ranks the examples perfectly, all goods come before bads, and the values of f are unchanged. As a result, the new probability estimate is 0 for all goods and 1 for all bads. PAV provides a set of intervals with a corresponding set of estimates $f(i)$ for every interval i. In order to use the results on new data, one must map a probability estimate to an observation according to its corresponding interval for which the score value is between the lower and upper boundaries (Zadrozny & Elkan, 2001). In order to apply IR on the uncalibrated GP output, a validation set is used to train the IR classifier using 10-fold cross-validation. The classifier is subsequently employed on the test set. The calibration plots in Fig. 12 illustrate the process of calibration on the training set in the IR plot. Also, a reliability curve quantifies how well the resulting set of isotonic values is calibrated. Finally, the figure shows the respective distribution of score values in a histogram. The IR diagram in Fig. 12 shows the process of calibration. Clearly visible are the intervals and their corresponding probability estimates to which the test set is mapped. The jump in the probability of default from 0.3 to 0.6 is noteworthy. Hence, small differences in score value lead

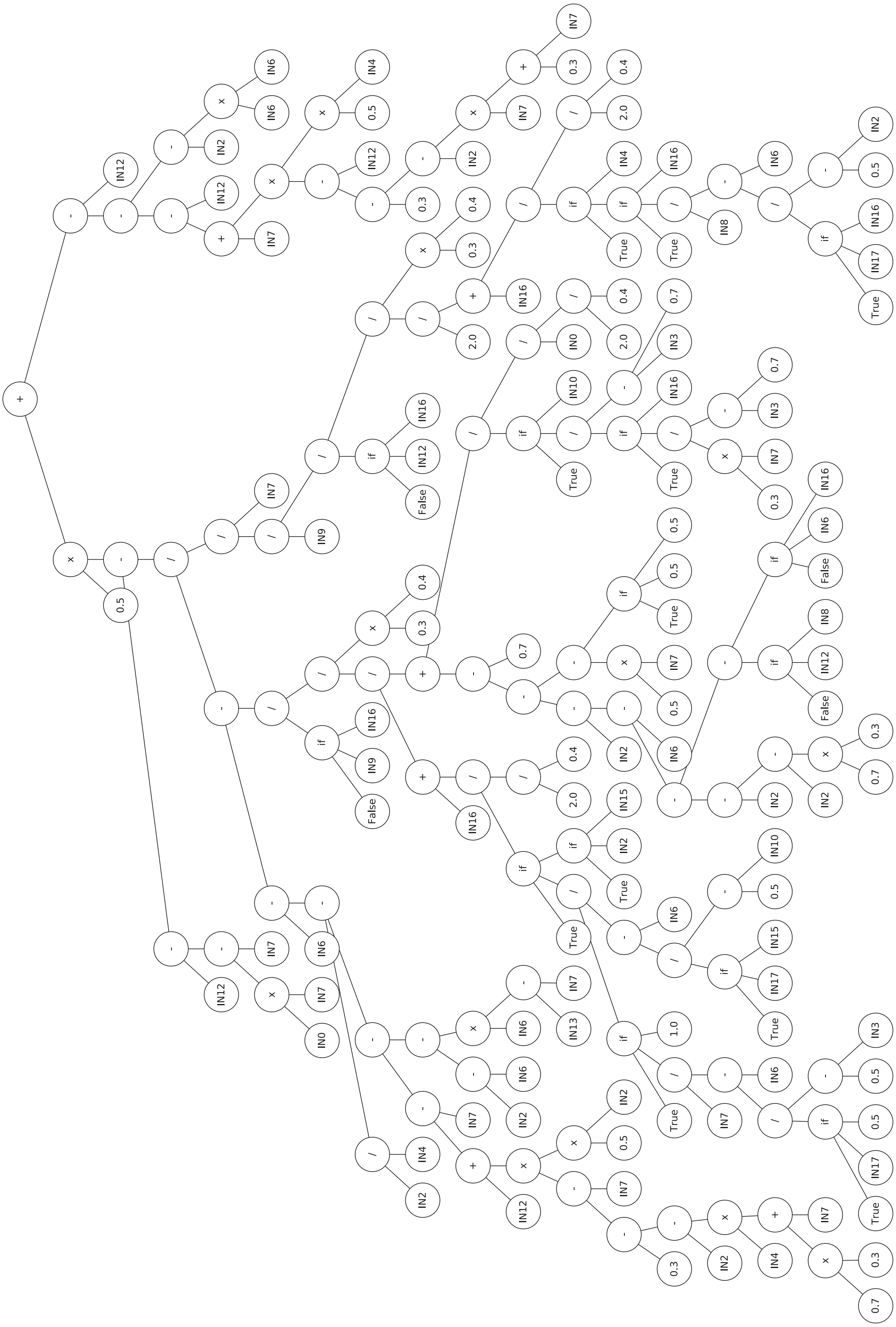

Fig. 11. Model with highest fitness value.

to large differences in the probability of default. Furthermore, the interval with the highest probability of default extends over a wide score range but for a comparatively low probability of default. The implication is that the GP model ranks incorrectly for high score values. This is consistent with Zadrozny and Elkan (2001), in which authors point out that, in the general case, PAV averages out more observations in score ranges where the scoring model ranks properly. Hence, the GP model works better in the score range for lower default probability.

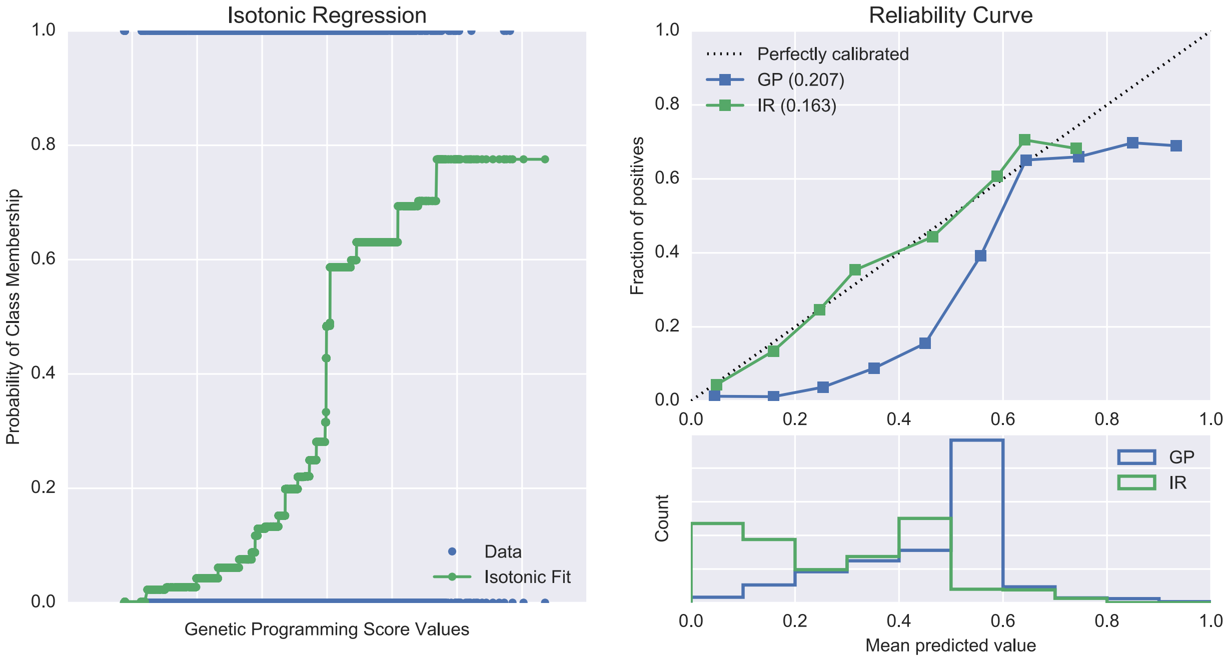

Fig. 12. Calibration plots.

The reliability diagram illustrates how well-calibrated the predicted probabilities from the normalized uncalibrated GP output and from the IR-calibrated GP probabilities are. For this purpose, the mean predicted values are plotted against the true fraction of positives. Due to the high number of observations, the score values are discretized into 10 bins. A perfectly calibrated classifier has the same fraction of positives as mean-predicted values for every class, which implies a score value corresponding to its assigned default probability. Such a classifier is represented by a diagonal line in the reliability plot (DeGroot & Fienberg, 1983; Zadrozny & Elkan, 2002). In addition, the Brier Score is used as a verification measure to assess the calibration accuracy of GP and IR-calibrated GP. The Brier Score is defined by the mean square error of the probability forecasts (see Eq. (4)), which means that a lower score corresponds to a higher accuracy (Brier, 1950). As we can see from Fig. 12, the GP output values are not well calibrated, and the use of IR remarkably improves the calibration's quality. The normalized uncalibrated GP output values follow a sigmoid-like shape below the perfectly calibrated diagonal line. Instead, after calibration with IR, the output values follow the diagonal of perfectly calibrated probabilities with only very small and slightly visible discrepancies. While the results of the IR calibration of the outputs of GP are in general remarkably good, it should also be pointed out that for the highest GP output values, the uncalibrated normalized output differs from the IR-calibrated output only marginally. Furthermore, the calibrated results are only distributed up to a 0.7 default probability, which is a result of the constant value for fractions of positives at around 0.7 for the four bins with the highest uncalibrated GP output values. Hence, the highest uncalibrated GP output values have an identical probability of default. The Brier Scores confirm the visual examination with an increase from 0.207 of the uncalibrated GP output to 0.163 of the IR-calibrated probabilities. The histogram of the GP output values in Fig. 12 shows that most of the observations are scored in the middle and lower areas of the score range. Furthermore, the score range with the highest number of observations is also better calibrated initially. After calibration, the GP output values are more consistently distributed in the lower score range but still incorporate few observations in the higher score range. The histogram further exhibits an inherent effect of IR: calibrated scores shift toward the tails of the score range. In conclusion, IR is applied to learn a mapping from ranking scores to calibrated probability estimates. Transforming scores using IR yields significant improvement, and the calibrated scores now translate to default probabilities.

Fig. 13 reports the distribution of goods and bads as well as the average default rate in their corresponding score ranges. As we can observe, goods and bads are clearly separated, with bads mainly in the low score range and goods mainly in the high score range. Furthermore, the arithmetic means of goods and bads are reasonably far from each other, as depicted by the black arrows. The lowest score groups have more than 70% bads while the highest score ranges contain few bads overall. Most bads are distributed among three score groups, which also incorporate a high number of goods. The default rate is steadily decreasing with increasing score values, which mirrors the calibration results shown in Fig. 12. Clearly visible is a decrease in score groups compared to the original pre-score in Fig. 7, which mirrors a decrease in distribution spread over the score range.

To sum up, GP visibly separates goods and bads but provides a decrease in score range. The model seems to better discriminate among goods than bads, providing some score groups with mainly goods but no score groups with only bads.

7.3. Discriminatory power of GP and comparison with other classifiers

In this section, the GP's performance is compared to that of three state-of-the-art machine learning methods, specifically logistic regression (LR), support vector machines (SVMs), and boosted trees (BTs).

To perform these experiments, we relied on the scikit-learn library (Pedregosa et al., 2011b), a machine learning tool in Python. For the SVM, the implementation provided in the class SupportVectorClassification (SVC) was used. The SVM is trained with the sequential minimal optimization (SMO) algorithm (Platt, 1998), and the parameters' values were selected after a preliminary tuning phase. In particular, for the kernel, we considered a polynomial, a linear, and a radial basis function kernel during the tun-

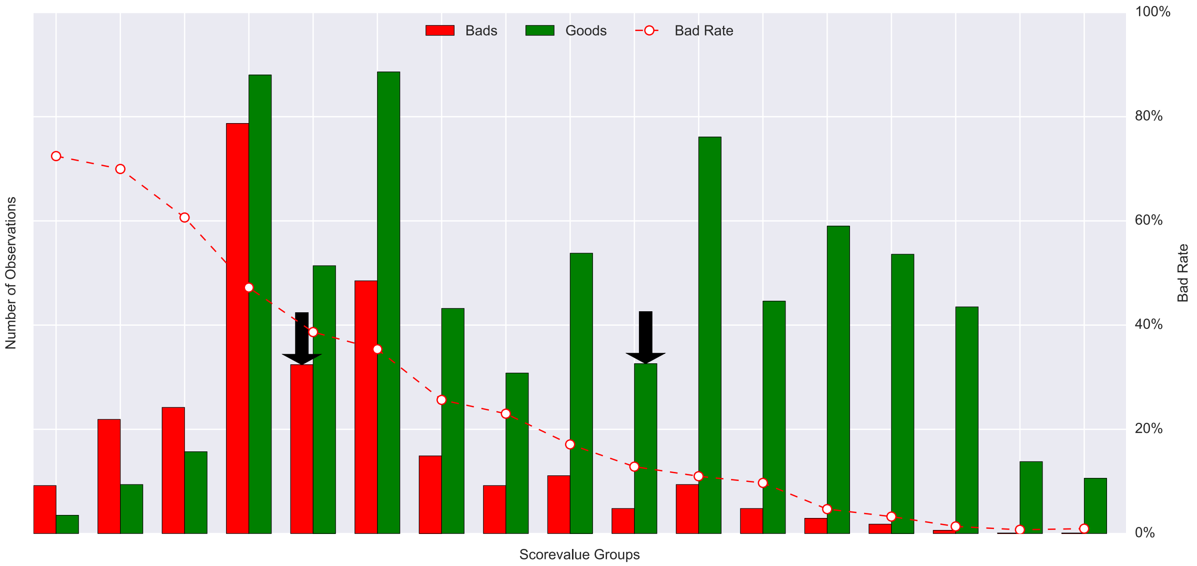

Fig. 13. GP distribution.

Table 5
Effect of IR on GP, SVM, and BT.

	Uncalibrated ROC-AUC		Calibrated ROC-AUC	Uncalibrated PR-AUC		Calibrated PR-AUC
GP	0.779	→	0.777	0.541	→	0.545
SVM	0.754	→	0.756	0.56	→	0.564
BT	0.78	→	0.779	0.54	→	0.54

ing phase, and we relied on the latter during the experiments. The maximum number of iterations was 100,000, probability estimates (Lin, Lin, & Weng, 2007) were enabled, and the shrinking heuristics (Joachims, 1998) were used to speed up the optimization. For the BT, the class AdaBoostClassifier was used. The class implements an AdaBoost (Rätsch, Onoda, & Müller, 2001) classifier, which is a meta-estimator that begins by fitting a classifier on the original dataset. Then, it fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted, such that subsequent classifiers focus more on difficult cases (Pedregosa et al., 2011b). After a preliminary tuning phase, we ended up with the following parameter values: the base estimator (classifier) from which the boosted ensemble is built consists of a decision tree classifier, the maximum number of estimators at which boosting is terminated was equal to 2,000, and the learning rate was 1. The SAMME.R real boosting algorithm (Hastie, Rosset, Zhu, & Zou, 2009a) was used for its faster convergence and better generalization ability (Hastie, Tibshirani, & Friedman, 2009b) with respect to SAMME discrete boosting algorithm. Similarly, GP, SVM ,and BT are subject to IR to ensure properly calibrated outputs. LR outputs well-calibrated predictions already, and thus IR is not applied; in fact, employing IR on well-calibrated methods does not improve the results but may hurt the performance (Niculescu-Mizil & Caruana, 2005). The effect of IR on ROC-AUC and PR-AUC is shown in Table 5. As we can see, the effect is only marginal and therefore negligible.

Fig. 14 shows the ROC curve and the PR diagram for all the studied methods, including the numeric values of the ROC-AUC and of the PR-AUC. As the figure shows, the GP model seems to perform reasonably well in the whole ROC space, with better performance in the ROC space's more liberal area, i.e., the area on the upper right-hand side. Liberal classifiers make positive classifications with weak evidence; hence, they classify a majority of bads correctly (high TPR) but also classify a high number of goods as bads (high FPR). In contrast, classifiers that are strong on the lower right-hand side are called conservative because they classify only with strong evidence (Fawcett, 2003). Subsequently, they have a small TPR but a small FPR, as well. A similar behavior occurs in the PR diagram. The precision for low recall values is rather low but drops comparably lightly with increasing recall values. This mirrors the distribution of goods and bads, as reported in Fig. 13. The lowest score group shows a bad rate of around 70%, which can be mapped to the precision in low recall values. Moving the threshold to include the three score groups with the most bads lowers the precision due to the decrease in the bads rate. Moving the threshold even further mainly rejects goods, leading to a rapid decrease of precision. Subsequently, GP shows a strong performance for high recall values.

Comparing the GP results to the other models reveals that the classifiers' overall performances are all similar. Nonetheless, some interesting differences can be observed. The LR model has a similar performance to the GP model in the liberal area. However, in the very conservative area, LR seems to be slightly better than GP while from the conservative area to the liberal area, the performance of LR is visibly worse. The ROC-AUC of the GP model (0.78) is higher than that of LR (0.76). In the PR diagram, LR shows a better precision for very low recall values and similar precision for very high recall values, compared to GP. In between the two, GP's precision is better than LR's, which overall leads to a nearly identical PR-AUC value, with 0.54 for GP and 0.53 for LR. The SVM model reveals a strong performance in the ROC diagram's conservative area, with a ROC curve visibly above the other models. However, when leaving the conservative area, SVM's curve falls below the other models' curves, and it remains lower throughout the diagram. Overall, the SVM's ROC-AUC is 0.76, which is worse than GP's ROC-AUC. Contrarily, SVM's performance in the PR diagram is the best, with a PR-AUC value of 0.56. The high performance in PR-AUC comes from the lower recall values, for which SVM shows a very high precision. However, from the central recall values forward, the precision drops drastically below the other classifiers' performances. The BT model shows a similar performance to GP's with a ROC-AUC value of 0.78. Also, in the PR diagram, BT shows a performance similar to GP's, with a PR-AUC value of 0.54. BT's precision is very high for low recall levels but drops rapidly for high recall values. However, the precision increases again and stays above the other models for the higher recall values.

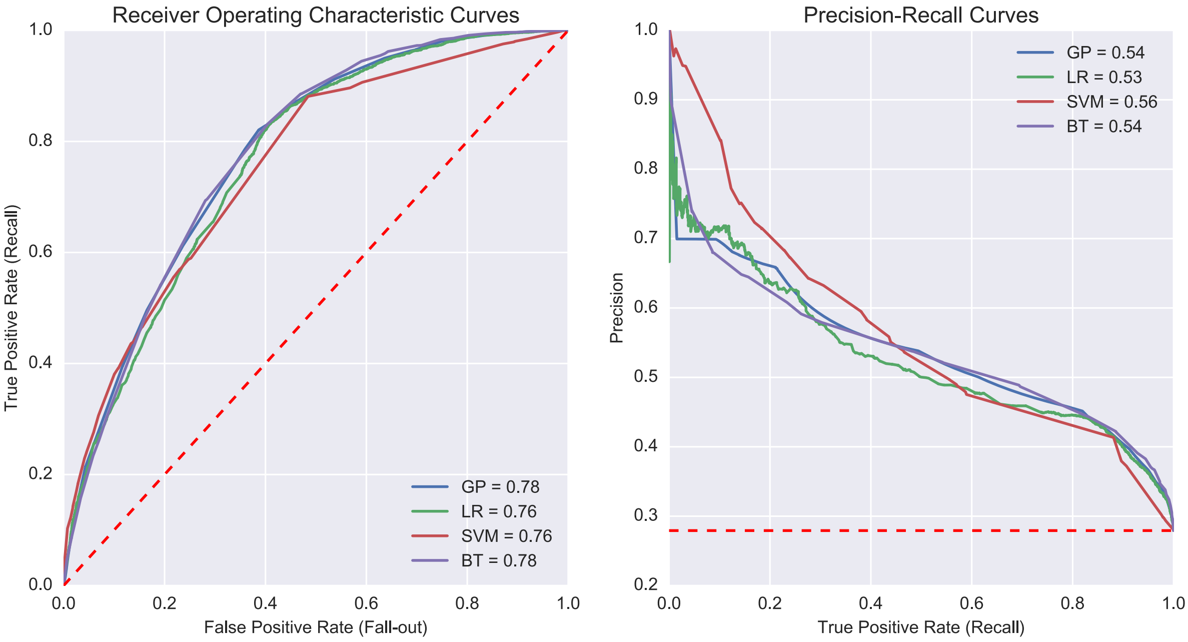

Fig. 14. ROC curve and PR diagram.

In conclusion, the analysis shows that while the models share similar overall performances, their performances in the different areas of the ROC space differ. GP shows a consistently good performance over the whole ROC space. LR performs reasonably well in the very liberal and very conservative areas but poorly in between. SVM has good performance in the conservative area and poor performance in the liberal area of the ROC space. Finally, BT has a poor performance in the conservative area and a good performance in the rest of the ROC space. A similar trend can be observed in the PR diagram. While the PR-AUC values do not vary much, the methods reveal different performances in various areas of the diagram. One can make an interesting observation from the SVM model: the performance in the ROC space is visibly worse than that of its competitors while the performance in the PR space is visibly better. For low recall values, the SVM model provides the most relevant results. On the other hand, the results become less precise for high recall values.

7.4. Discriminatory power of GP in collaboration with the credit agency score

While the pre-score needs to be operational on its own in countries that do not or cannot offer a credit agency score based on country-specific solvency information, an additional requirement is the usability in combination with the main score. Here, as in the main score, we consider the AFS company's already-existing credit agency score, also commonly named ABIC, or reference score. AFS currently uses this score. The pre-score needs to have an enhancing effect on the main score's discriminatory power. For this purpose, both scores are combined by considering their arithmetic means. Then the effect of the scores' combination is analyzed. Fig. 15 shows the score distribution of the combination of ABIC and GP. The discriminatory capabilities are clearly visible. The bads (represented by red bars) are more numerous on the low score value range than the goods (represented by green bars). However, the bads are spread out rather evenly, with a visually regular, Gaussian-like shape. Contrarily, the goods are strongly centered around their mean value, with a smaller spread toward the outer score groups. The bad rate, reported using a dashed red line, shows high bad rates between 80% and 100% for the score range's first quarter and very low bad rates, i.e., below 5%, for the score range's fourth quarter. Correspondingly, the bads rate drops drastically from around 80% to around 5% within the score range's second and third quarters. Compared to GP, ABIC + GP incorporates a greater score range with better-distributed scores. The bads rate in the low score range is higher and approaches score groups with mainly bads. Similar to GP, the bads rate in the high score range is near zero. Consequently, the score groups in the high score range mainly consist of goods.

7.4.1. Comparison between GP, ABIC, ABIC+GP and pre-score with varying threshold

Fig. 16 shows a comparison between the discriminatory powers of GP, ABIC, and ABIC+GP and the original pre-score. The original pre-score, which is clearly outperformed by GP, is especially weak in the ROC space's more conservative areas but gains some discriminative power in the most liberal area. While its ROC curve has a similar shape to GP's curve for most of the ROC space, it also lies constantly below GP's curve and only approaches it in the very liberal area. A similar trend also appears in the PR diagram. While the original pre-score's PR curve approaches the GP curve for both very low and very high recall values, its curve is steadily lower than the GP curve. Furthermore, the pre-score's PR curve reveals a clear concave shape while every other classifier displays a more convex shape, a clear indication that the pre-score is a weaker classifier than the others. ABIC shows a performance comparable to GP's in the ROC diagram's conservative area but drops drastically in performance in the diagram's liberal area, where it is clearly outperformed by GP. The ROC-AUC for ABIC is low (0.71). However, combining GP and ABIC boosts the performance of both considerably to a ROC-AUC of 0.81, which is a remarkably better value than that of all the other studied models. Also, one can easily observe that the ROC curve of ABIC+GP is high in both conservative and liberal areas, outperforming all the other methods and perfectly integrating the contradictory curve movements of ABIC and GP when considered in isolation. Additionally, while both scores perform comparably in the conservative area, combining both scores still boosts their performances. Thus, the scores do not identify the same bads but different ones. The PR diagram shows a similar picture. ABIC shows high precision in the lower recall area but drops over the

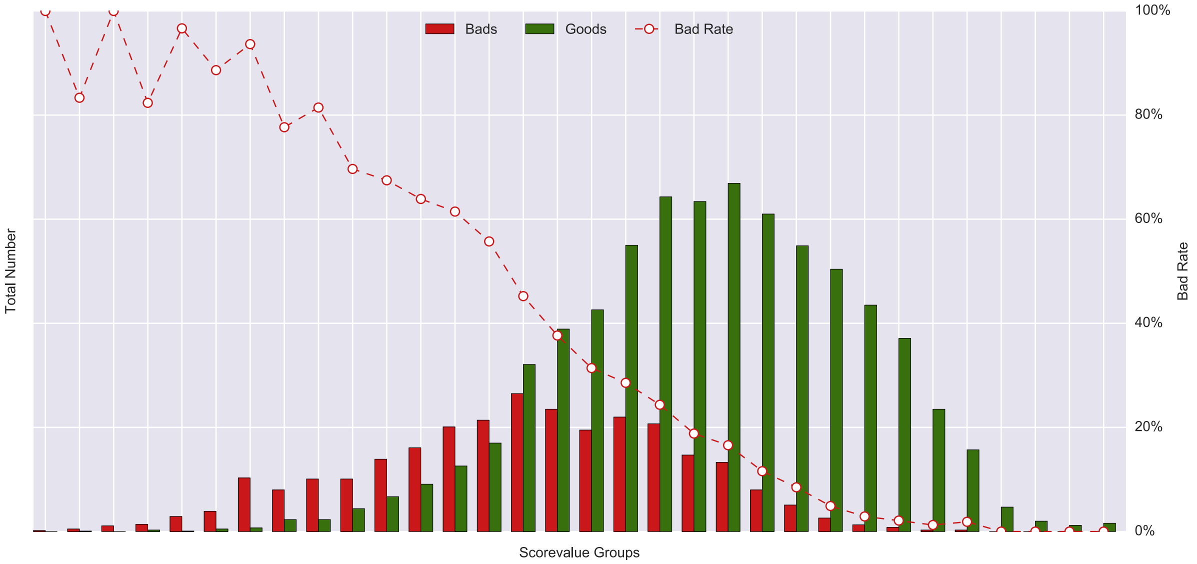

Fig. 15. Score distribution of ABIC and GP.

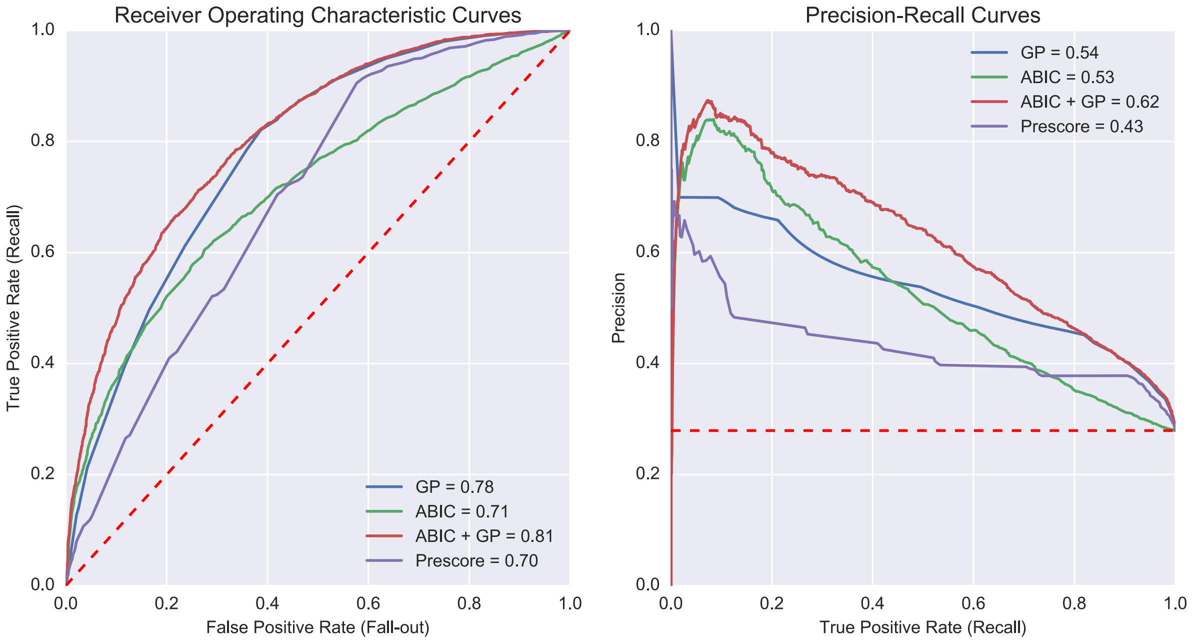

Fig. 16. ROC curve and PR diagram for pre-score, ABIC, GP, and GP integrated with ABIC (ABIC+GP).

recall range. Consequently, ABIC shows the lowest precision in the high recall ranges and then drops below GP. On the other hand, GP has a high precision in the high recall area. The combination of ABIC and GP visibly outperforms the other methods in the PR diagram.

In conclusion, the analysis of the ROC and PR diagrams visibly demonstrates the lack of discriminatory power in the pre-score, thus incentivizing this work. Furthermore, GP demonstrated its capabilities in separating bads from goods, and when used in combination with the credit agency score, it was clearly able to outperform all other methods.

7.4.2. Experimental results with fixed threshold

The ROC and PR diagrams allow comparison of the discriminative powers of binary classifiers, as their discrimination thresholds are varied. However, the discrimination threshold is essential for a classifier's application. A number of various methods for quantifying prediction quality at a specific threshold exist. The method used most frequently in research is possibly accuracy (ACC), i.e., the fraction of classifications that are correct (Davis et al., 1992; García, Marqués, & Sánchez, 2014; Zhang et al., 2007). One possible way to find the optimal threshold consists of calculating the accuracy for every possible threshold and then returning the threshold with the highest accuracy. Table 6 shows the highest accuracy for GP, pre-score, ABIC and ABIC+GP. The accuracy results are consistent with the previous results: GP and ABIC have comparable accuracies; together, they form a more powerful classifier with the highest accuracy (78.5%). The pre-score obtains a weaker accuracy that the other methods (72.8%).

While accuracy is one of the most used classification evaluation criteria, it is biased for the majority class and therefore not recommended for unbalanced datasets (García et al., 2014). Additionally, accuracy and many other evaluation techniques assume symmetrical misclassification costs for goods and bads. Often, however, the

Table 6
Classifier accuracies of GP, pre-score, ABIC, and ABIC+GP. Highest values of the accuracy obtained for all possible values of the thresholds for each method are reported.

Classifier	Accuracy
GP	75.0%
pre-score	72.8%
ABIC	75.9%
ABIC + GP	78.5%

costs of accepting bads are higher than the costs of rejecting goods. Consequently, instead of using an evaluation technique based on the overall error, one should employ a cost function (Frydman, Altman, & Kao, 1985; West, 2000). If C_1 denotes the cost of accepting bads, C_2 denotes the cost of rejecting goods and π_1 and π_2 are the ratios of goods and bads in the population, respectively, then, following Lee and Chen (2005), the cost function is defined as

$$\text{Cost} = C_1 * \text{FNR} * \pi_1 + C_2 * \text{FPR} * \pi_2$$

While this cost function takes into account the misclassification costs for goods and bads, it assumes constant order values across classes. Because the order values used for the dataset are known, a profit-based evaluation technique is employed in which order values are adjusted to include misclassification costs. The misclassification costs are defined as abortion rate and marginal return and are provided by AFS. All requests classified as goods are reduced to 25% of their original value, which corresponds to the profit margin. Additionally, requests for goods classified as bads (FP) are reduced by another 30%, which corresponds to the abortion rate of potential customers who are unwilling to pay via the offered payment types. Bads that are classified as goods (FN) are reduced by the profit margin and deducted from the sum of the remainder, excluding bads that are classified as bads (TP) and subsequently rejected. In the latter case, it is assumed that rejected bads do not continue the order process. The calculations are reported in Eqs. (5) through (9).

$$\text{TP} = 0 \tag{5}$$

$$\text{FN} = \text{Ordervalue} * (1 - \text{Profit Margin}) \tag{6}$$

$$\text{FP} = \text{Ordervalue} * \text{Profit Margin} * (1 - \text{Abortion Rate}) \tag{7}$$

$$\text{TN} = \text{Ordervalue} * \text{Profit Margin} \tag{8}$$

$$\text{OV} = \text{TN} + \text{FP} - \text{FN} \tag{9}$$

The bads have a mean order value that is 2% higher than the mean order value of the goods. However, taking into account the misclassification costs, the values change considerably. Using TN as the determination base yields FN that is 206% higher than TN and FP that is 30% lower (=abortion rate) than TN. This means that 1 euro of profit for goods classified as goods yields 0.7 euros of profit if they are classified as bads, instead. Similarly, if they are classified as goods but turn out to be bads, they yield a loss of 3.06 euros. Hence, falsely changing the classification from good to bad induces profit deduction of $1 - 0.7 = 0.3$. Falsely keeping the classification as good induces profit deduction of 3.06 euros. Accordingly, for every falsely classified bad, 10 goods can be falsely classified as bads. Results of GP, pre-score, ABIC and ABIC+GP are depicted in Table 7, where a pessimistic classifier that rejects nothing (PES) and an optimal classifier that rejects only bads (OPT) are compared to one another. In order to allow for interclass analysis, the results for bads and goods are shown as supplementary to the totals. Additionally, the accuracy is included for comparison purposes. The pessimistic classifier accepts all bads and therefore has a high bads deduction on the overall order value. At the same time, the order value for goods is at a maximum because no goods are rejected. The results show a strong decrease in bads deduction (increase in OV) for all models, with GP being better than the original pre-score. Furthermore, while ABIC has the highest order value decrease for goods, GP has the lowest, which leads to a moderate order value decrease for ABIC+GP. Due to the imbalanced dataset, the impact of goods is higher than the impact of bads. Subsequently, ABIC has the worst performance as a result of the high order value deduction for goods. Nonetheless, ABIC+GP performs similarly to GP, sharing the highest increase in order value compared to the pessimistic classifier.

These results contrast with the accuracy results, in which the pre-score performs worst, outperformed by GP and ABIC, which in turn are outperformed by ABIC+GP. Also, unlike the order value, the accuracy shows little deviation between the pessimistic classifier and the models, with a minimum value of 1.9% for the pre-score and a maximal value of 8.6% for ABIC+GP. On the other hand, comparing the models with the optimal value shows high deviations for accuracy but little for order value. The optimal classifier discriminates perfectly between goods and bads and therefore rejects all bads and accepts all goods. Consequently, there is no order value deduction for bads, and the order value for goods is maximal. The results show the differences between accuracy and order value as an evaluation measure: accuracy shows the models to be rather similar to a pessimistic classifier while the order value shows the models to be rather similar to the optimal classifier. Table 8 shows the difference in order value between GP and the other considered methods, i.e., pre-score, ABIC and ABIC+GP.

Consistent with the previous results, the pre-score's order values are lower than GP's values, with −15% for the bads, −1.6% for the goods and a total of −2.4%. ABIC+GP again offers the best results: the order value of bads is 30.9% higher for ABIC+GP than for GP, but the order value for goods is 1.4% smaller. The impact of the smaller order value for goods is higher than the impact of the higher order value for bads because the dataset is unbalanced. Hence, the total difference adds up to only a 0.1% increase from GP to ABIC+GP. To interpret these results, one has to observe the ABIC results. In fact, for ABIC, the highest order value threshold is found on a level that effectively rejects all bads and many goods. Consequently, there is no discount from bads, but one can observe a high difference in the GP results for goods (−10.3%), instead. Hence, ABIC has a total difference of −5.7%.

8. Conclusions and future work

The objective of this work was to develop a credit scoring (CS) model to replace the pre-risk check of the e-commerce risk management system Risk Solution Services (RSS), which is currently one of the most used systems to estimate customers' default probabilities. The pre-risk check uses data from the order process and includes exclusion rules and a generic CS model. The new model was supposed to work as a replacement for the whole pre-score and had to be able to work in isolation and in integration with the RSS main risk check. The focus of the paper was developing a model based on genetic programming (GP) to predict the probability of default on payment. The presented results have shown a profit increase of around 18.6% by employing a CS model based on GP, compared to not employing CS at all. In order to evaluate the discriminatory power of the GP model, this model was compared to models based on logistic regression, support vector machines, and boosted trees, using ROC analysis. Even though the fitness function used by our GP system was the area under the

Table 7
Order value comparison among GP, pre-score, ABIC, and ABIC+GP.

		GP		Pre-score		ABIC		ABIC + GP	
		OV	ACC	OV	ACC	OV	ACC	OV	ACC
PES	Bads	22.4%	2.9%	22.3%	2.1%	23.0%	5.5%	22.6%	9.0%
	Goods	−3.8%	−0.2%	−4.0%	−0.2%	−5.2%	−0.2%	−4.0%	−0.4%
	Total	18.6%	2.7%	18.3%	1.9%	17.9%	5.2%	18.6%	8.6%
OPT	Bads	−0.7%	−20.2%	−0.8%	−21.0%	0.0%	−17.6%	−0.5%	−14.1%
	Goods	−3.8%	−0.2%	−4.0%	−0.2%	−5.2%	−0.2%	−4.0%	−0.4%
	Total	−4.5%	−20.3%	−4.8%	−21.2%	−5.2%	−17.8%	−4.4%	−14.5%

Table 8
Deviation from GP of the other studied methods.

	Pre-score	ABIC	ABIC+GP
Bads	−15.0%	100%	30.9%
Goods	−1.6%	−10.3%	−1.4%
Total	−2.4%	−5.7%	0.1%

ROC curve (ROC-AUC), the PR diagram (and the area under this diagram, PR-AUC) was also used to compare the various methods' performances. The GP model shows a higher discriminatory power than the pre-score when ROC and PR are used and when using error-based evaluation techniques. GP was evaluated not only with error-based evaluation techniques but also with profit. Analyzing the profit obtained by the GP model, we can see that this model shows an increase of 2.4% over the pre-score. Combining GP and the credit agency score (ABIC) into a single score increases its predictive power. In ROC, PR, and accuracy analysis, ABIC and GP's integration shows the highest performance among all the studied methods. On the basis of these empirical results, one can conclude that GP can generate models with higher discriminatory power than the pre-score and that it is competitive with other state-of-the-art machine learning systems. Also, GP works particularly well when integrated with the existing credit agency score. Therefore, the GP-generated models can replace and improve the pre-score.

This research can be extended in a number of ways. Most importantly, the lack of variables has to be addressed. Although it is an obvious proposal to increase a model's discriminatory power by adding variables with new information value, it is not without clear reasoning. The fact that the proposed model is designed with fewer variables than its predecessor and the fact that the AFS company's analysts identified the missing variables as powerful illustrate the importance. Furthermore, the model complexity and interpretability issue in GP is worth considering for future research. In fact, on one hand, it is true that GP models are expressed in the form of a tree, which adds some transparency, like the fact that the model can be reported for future analysis, as in Fig. 11. However, it is also undeniable that unless the tree is small, it is often unreadable and very hard, or even impossible, to understand. On the other hand, if the tree becomes too small, the estimates obtained from it could have discontinuities, thus making them not appealing in practice. Another possible improvement of the current work would be using more sophisticated GP systems than the standard GP used here. For instance, it is appropriate to apply recently introduced GP systems that integrate semantic awareness in the evolution toward improved search performance. The interested reader is referred to Castelli, Silva, and Vanneschi (2015), Vanneschi (2017) and Ruberto, Vanneschi, Castelli, and Silva (2014) for an introduction to the most popular of these GP systems.

Appendix A. An introduction to genetic programming

Genetic programming (GP) (Koza, 1992b) is a method that belongs to the computational intelligence research field called evolutionary computation (Eiben & Smith, 2003). GP consists of the automated learning of computer programs by means of a process inspired by Charles Darwin's theory of biological evolution. In the context of GP, one can interpret the word *program* in general terms, and therefore, GP can be applied to the particular cases of learning expressions, functions and, as in this work, data-driven predictive models. In GP, programs are typically encoded by defining a set, $\mathcal{F}$, of primitive functional operators and a set, $\mathcal{T}$, of terminal symbols. Typical examples of primitive functional operators may include arithmetic operations (+ , - , *, etc.); other mathematical functions (such as sin, cos, log, exp) or, according to the context and type of problem, Boolean operations (such as AND, OR, NOT) or more complex constructs such as conditional operations (such as If-Then-Else); iterative operations (such as While-Do); and other domain-specific functions that may be defined. Each terminal is typically either a variable or a constant, defined in the problem domain. GP's objective is to navigate the space of all possible programs that can be constructed by composing symbols in $\mathcal{F}$ and $\mathcal{T}$, looking for the most appropriate ones for solving the problem at hand. Generation by generation, GP stochastically transforms populations of programs into new, hopefully improved program populations. The appropriateness of a solution in solving the problem (i.e., its quality) is expressed by using an objective function (the fitness function). In order to transform a population into a new population of candidate solutions, GP selects the most promising programs that are contained in the current population and applies to those programs some particular search operators called genetic operators, typically crossover and mutation. The standard genetic operators (Koza, 1992b) act on the structure of the programs that represent the candidate solutions. In other terms, standard genetic operators act at a syntactic level. More specifically, standard crossover is traditionally used to combine two parents' genetic material by swapping a part of one parent with a part of the other. Considering the standard tree-based representation of programs often used by GP (Koza, 1992b), after choosing two individuals based on their fitness, standard crossover selects a random subtree in each parent and swaps the selected subtrees between the two parents, thus generating new programs (the offspring). On the other hand, standard mutation introduces random changes in the structures of the population's individuals. For instance, the traditional and most-frequently used mutation operator, sub-tree mutation, works by randomly selecting a point in a tree, removing whatever is currently at and below the selected point and inserting a randomly generated tree at that point. The reader who is interested in more details is referred to Poli, Langdon, and Mcphee (2008b) and Koza (1992b).

References

Abdou, H. A. (2009). Genetic programming for credit scoring: The case of Egyptian public sector banks. *Expert Systems with Applications, 36*(9), 11402–11417.

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intelligent Systems in Accounting, Finance & Management, 18*(2–3), 59–88.

Abelln, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert*

Systems with Applications, 41(8), 3825–3830. doi:10.1016/j.eswa.2013.12.003. http://www.sciencedirect.com/science/article/pii/S0957417413009676

Alves, B. C., & Dias, J. G. (2015). Survival mixture models in behavioral scoring. *Expert Systems with Applications, 42*(8), 3902–3910. doi:10.1016/j.eswa.2014.12.036. http://www.sciencedirect.com/science/article/pii/S095741741400815X

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., Silverman, E., Rpeid, W. T., et al. (1955). An empirical distribution function for sampling with incomplete information. *Source: The Annals of Mathematical Statistics, 26219247*(4), 641–647.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J. A. K., & Vanthienen, J. (2003). Benchmarking state of the art classification algorithms for credit scoring. *Journal of the Operational Research Society*.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1–3.

Castelli, M., Silva, S., & Vanneschi, L. (2015). A c++ framework for geometric semantic genetic programming. *Genetic Programming and Evolvable Machines, 16*(1), 73–81.

Code, U. S. (1974). Equal credit opportunity act.

Fittkau & Maa Consulting (2014). 38. WWW-Benutzer-Analyse W3B: Kaufentscheidung im internet.

Davis, R. H., Edelman, D. B., & Gammerman, A. J. (1992). Machine-learning algorithms for credit-card applications. *IMA Journal of Management Mathematics, 4*(1), 43–51.

DeGroot, M. H., & Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *The Statistician, 32*(1), 12–22.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. SpringerVerlag.

Fawcett, T. (2003). ROC graphs : Notes and practical considerations for data mining researchers. *HP Invent*, 27.

Fitzpatrick, T., & Mues, C. (2016). An empirical comparison of classification algorithms for mortgage default prediction: Evidence from a distressed mortgage market. *European Journal of Operational Research, 249*(2), 427–439. doi:10.1016/j.ejor.2015.09.014. http://www.sciencedirect.com/science/article/pii/S0377221715008383.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*, 2171–2175.

Frigge, D. (2016). Online-payment 2016.

Frydman, H., Altman, E. I., & Kao, D.-L. (1985). Introducing recursive partitioning for financial classification: The case of financial distress. *The Journal of Finance, 40*(1), 269–291.

García, V., Marqués, A. I., & Sánchez, J. S. (2014). An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *Journal of Intelligent Information Systems, 44*(1), 159–189.

Giovanni, S., & Elder, J. F. (2010). Ensemble methods in data mining: Improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery, 2*(1), 1–126.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc..

Han Ju, Y., & Young Sohn, S. (2014). Updating a credit-scoring model based on new attributes without realization of actual data, 234, 119126.

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.

Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009a). Multi-class adaboost. *Statistics and its Interface, 2*(3), 349–360.

Hastie, T., Tibshirani, R., & Friedman, J. (2009b). Unsupervised learning. In *The elements of statistical learning* (pp. 485–585). Springer.

Henley, W. E. (1995). *Statistical aspects of credit scoring*. Open University Ph.D. thesis..

Henley, W. E., & Hand, D. J. (1996). A k-nearest-neighbour classifier for assessing consumer credit risk. *The Statistician, 45*(1), 77.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Huang, J. J., Tzeng, G. H., & Ong, C. S. (2006). Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation, 174*(2), 1039–1053.

Iba, H., Garis, H. D., & Sato, T. (1994). Genetic programming using a minimum description length principle. *Advances in Genetic Programming*, 265–284.

Joachims, T. (1998). Making large-scale SVM learning practical. *Technical Report*. Technical report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Kirschen, R. H., O'Higgins, E. A., & Lee, R. T. (2000). The royal london space planning: An integration of space analysis and treatment planning. *American Journal of Orthodontics and Dentofacial Orthopedics, 118*(4), 448–455.

Koza, J. R. (1992a). *Genetic programming: On the programming of computers by means of natural selection*: vol. 1. MIT press.

Koza, J. R. (1992b). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press.

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning,. 40, 5125–5131.

Lahsasna, A., Ainon, R. N., & Wah, T. Y. (2010). Credit scoring models using soft computing methods: a survey. *The International Arab Journal of Information Technology, 7*(2), 115–123.

Lee, T., & Chen, I. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*(4), 743–752.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research,. doi:10.1016/j.ejor.2015.05.030.

Li, Z., Tian, Y., Li, K., Zhou, F., & Yang, W. (2017). Reject inference in credit scoring using semi-supervised support vector machines. *Expert Systems with Applications, 74*, 105–114. doi:10.1016/j.eswa.2017.01.011. http://www.sciencedirect.com/science/article/pii/S095741741730012X.

Lin, H.-T., Lin, C.-J., & Weng, R. C. (2007). A note on platts probabilistic outputs for support vector machines. *Machine learning, 68*(3), 267–276.

Luo, S., Kong, X., & Nie, T. (2016). Spline based survival model for credit risk modeling. *European Journal of Operational Research, 253*(3), 869–879. doi:10.1016/j.ejor.2016.02.050. http://www.sciencedirect.com/science/article/pii/S0377221716301035.

Malhotra, R., & Malhotra, D. K. (2002). Differentiating between good credits and bad credits using neuro-fuzzy systems. *European Journal of Operational Research, 136*(1), 190–211.

Marques, a. I., Garcia, V., & Sanchez, J. S. (2013). A literature review on the application of evolutionary computing to credit scoring. *Journal of the Operational Research Society, 64*(9), 1384–1399.

Mays, E. (2001). *Handbook of credit scoring*. Global Professional Publish.

Mester, L. J. (1997). What's the point of credit scoring ? *Business Review, 3*, 3–16.

Myers, J. H., & Forgy, E. W. (1963). The development of numerical credit evaluation systems. *Journal of the American Statistical Association, 58*(303), 799–806.

Neto, R., Adeodato, P. J., & Salgado, A. C. (2017). A framework for data transformation in credit behavioral scoring applications based on model driven development. *Expert Systems with Applications, 72*, 293–305. doi:10.1016/j.eswa.2016.10.059. http://www.sciencedirect.com/science/article/pii/S0957417416306145.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on machine learning - ICML '05 1999* (pp. 625–632). New York, New York, USA: ACM Press.

Ong, C., Huang, J., & Tzeng, G. (2005). Building credit scoring models using genetic programming. *Expert Systems with Applications, 29*(1), 41–47.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011a). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011b). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines,.

Poli, R., Langdon, W. B., & McPhee, N. F. (2008a). *A field guide to genetic programing*. Lulu Enterprises.

Poli, R., Langdon, W. B., & Mcphee, N. F. (2008b). A field guide to genetic programming. http://www.gp-field-guide.org.uk.

Rätsch, G., Onoda, T., & Müller, K.-R. (2001). Soft margins for adaboost. *Machine learning, 42*(3), 287–320.

Reichert, A. K., Cho, C.-C., & Wagner, G. M. (1983). An examination of the conceptual issues involved in developing credit-scoring models. *Journal of Business & Economic Statistics, 1*(2), 101–114.

Ruberto, S., Vanneschi, L., Castelli, M., & Silva, S. (2014). Esagp – a semantic gp framework based on alignment in the error space. In M. Nicolau, K. Krawiec, M. I. Heywood, M. Castelli, P. García-Sánchez, J. J. Merelo, V. M. Rivas Santos, & K. Sim (Eds.), *Genetic programming: 17th european conference, EuroGP 2014, Granada, Spain, April 23–25, 2014, Revised Selected Papers* (pp. 150–161)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sackmann, S., Siegl, M., & Weber, D. (2011). Ein ansatz zur verbesserung der steuerung des zahlungsausfallrisikos im E-Commerce (B-to-C). *Zeitschrift für Betriebswirtschaft, 81*(2), 139–153.

Seidenschwarz, H., Weinfurtner, S., Stahl, E., & Wittmann, G. (2014). Gesamtkosten von Zahlungsverfahren - Was kostet das Bezahlen im Internet wirklich?

Siddiqi, N. (2006). *Credit risk scorecards: Developing and implementing intelligent credit scoring* (3rd). John Wiley & Sons.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149–172.

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit scoring and its applications*. Society for Industrial and Applied Mathematics.

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. doi:10.1016/j.ejor.2011.10.007. http://www.sciencedirect.com/science/article/pii/S0377221711009064.

Vanneschi, L. (2017). An introduction to geometric semantic genetic programming. In O. Schütze, L. Trujillo, P. Legrand, & Y. Maldonado (Eds.), *NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop NEO 2015 held at September 23–25 2015 in Tijuana, Mexico* (pp. 3–42)). Cham: Springer International Publishing.

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research, 238*(2), 505–513. doi:10.1016/j.ejor.2014.04.001. http://www.sciencedirect.com/science/article/pii/S0377221714003105.

Wach, E. P. (2011). Trends im eCommere 2011.

Weinfurner, S., Weisheit, S., Wittmann, G., Stahl, E., & Pur, S. (2011). Zahlungsabwicklung im E-Commerce.

West, D. (2000). Neural network credit scoring models. *Computers and Operations Research, 27*(11–12), 1131–1152.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *The Journal of Financial and Quantitative Analysis, 15*(3), 757.

Yang, Z., Wang, Y., Bai, Y., & Zhang, X. (2004). Measuring Scorecard Performance. In M. Bubak, G. D. van Albada, P. M. A. Sloot, & J. Dongarra (Eds.), *Computational science ICCS 2004: vol. 3038* (pp. 900–906). Kraków: Springer-Verlag Berlin Heidelberg.

Yao, X., Crook, J., & Andreeva, G. (2015). Support vector regression for loss given default modelling. *European Journal of Operational Research, 240*(2), 528–538. doi:10.1016/j.ejor.2014.06.043. http://www.sciencedirect.com/science/article/pii/S0377221714005463.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. *Icml*, 1–8.

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining KDD 02*, 694–699.

Zhang, D., Huang, H., Chen, Q., & Jiang, Y. (2007). A comparison study of credit scoring models. *Proceedings - Third International Conference on Natural Computation, ICNC 2007, 1*(Icnc), 15–18.