

An Entropy Based Algorithm for Credit Scoring

Roberto Saia and Salvatore Carta

Dipartimento di Matematica e Informatica - Università di Cagliari
Via Ospedale 72 - 09124 Cagliari, Italy
{roberto.saia, salvatore}@unica.it

Abstract. The request of effective credit scoring models is rising in these last decades, due to the increase of consumer lending. Their objective is to divide the loan applicants into two classes, reliable or unreliable, on the basis of the available information. The linear discriminant analysis is one of the most common techniques used to define these models, although this simple parametric statistical method does not overcome some problems, the most important of which is the imbalanced distribution of data by classes. It happens since the number of default cases is much smaller than that of non-default ones, a scenario that reduces the effectiveness of the machine learning approaches, e.g., neural networks and random forests. The Difference in Maximum Entropy (DME) approach proposed in this paper leads toward two interesting results: on the one hand, it evaluates the new loan applications in terms of maximum entropy difference between their features and those of the non-default past cases, using for the model training only these last cases, overcoming the imbalanced learning issue; on the other hand, it operates proactively, overcoming the cold-start problem. Our model has been evaluated by using two real-world datasets with an imbalanced distribution of data, comparing its performance to that of the most performing state-of-the-art approach: random forests.

Keywords: Business intelligence · Credit scoring · Data mining · Classification

1 Introduction

The processes taken into account in this paper typically start with a loan application (from now on named as *instance*) and end with a repayment (or not repayment) of the loan. Although the retail lending represents one of the most profitable source of income for the financial operators, the increase of loans is directly related to the increase of the number of defaulted cases, i.e., fully or partially not repaid loans. In short, the *credit scoring* is used to classify, on the basis of the available information, the loan applicants into two classes, *reliable* or *unreliable* (or better, referring to their instances, *accepted* or *rejected*). Considering its capability to reduce the losses of money, it is clear that it represents an important tool, as stated in [1]. More formally, the *credit scoring* techniques can be defined as a group of statistical methods used to infer the probability that an instance leads toward a default [2,3].

Whereas that their processes involve all the factors that contribute to determine the *credit risk* [4] (i.e., probability of loss from a debtor's default), they allow the financial operators to evaluate this aspect. Other advantages related to these techniques are the

reduction of the credit analysis cost, a quick response time in the credit decisions, and the possibility to accurately monitor the credit activities [5]. The design of effective approaches for *credit scoring* is not a simple task, due to a series of problems, the most important of which is the imbalanced distribution of data [6] used to train the models (small number of default cases, compared to the non-default ones), which reduces the effectiveness of the machine learning strategies [7].

The idea behind this paper is to *evaluate an instance in terms of its features' entropy*, and to *define a metric able to determinate its level of reliability*, on the basis of this criterion. In more detail, we measure the difference, in terms of maximum *Shannon entropy* (from now on referred simply as *entropy*), between the same instance features, before and after we added to the set of non-default past instances an instance to evaluate. In the *information theory*, the entropy gives a measure of the uncertainty of a random variable. The larger it is, the less a-priori information one has on the value of it, then the entropy increases as the data becomes equally probable and decreases when their chances are unbalanced. It should be observed that, when all data have the same probability, we achieve the maximum uncertainty in the process of prediction of future data.

On the basis of the previous considerations, we evaluate a new instance in terms of uncertainty of its feature values: comparing the entropy of the set of non-default past instances, before and after we added to it the instance to evaluate. A larger entropy will indicate that the new instance contains similar data (increasing the level of equiprobability), otherwise, it contains different data, thus it represents a potential default case (in terms of non-similarity with the values of the non-default cases).

In our approach we choose to operate without take into account the default cases. Such strategy presents a twofold advantage: first, we can operate proactively, i.e., without the need to use default cases to train our model; second, we overcome the cold-start problem related with the scarcity (or total absence) of default cases, considering that in a real-world context they are much less than the non-default ones.

Given that in most of the cases reported in the literature [8,9,10] the *Random Forests* approach outperforms the other ones in this context, we will compare the proposed approach only to this one.

The main contributions of our work to the state of the art are listed below:

- (i) calculation of the *local maximum entropy by features* (Λ), which gives us information about the entropy achieved by each feature in the set of non-default cases (it allows us to measure the differences between instances in terms of single features);
- (ii) calculation of the *global maximum entropy* (γ), which represents a meta-feature based on the integral of the area under curve of the Λ values (it allows us to measure the difference between instances in terms of all features);
- (iii) formalization of the *Difference in Maximum Entropy (DME)* approach, used to classify the unevaluated instances as *accepted* or *rejected*, by exploiting the Λ and γ information;
- (iv) evaluation of the *DME* approach on two real-world datasets, by comparing its performance with those of a state-of-the-art approach such as *Random Forest* (in our case, without using the past default cases to train the model).

The remainder of the paper is organized as follows: Section 2 discusses the background and related work; Section 3 provides a formal notation and defines the problem faced in this paper; Section 4 describes the implementation of the proposed approach; Section 5 provides details on the experimental environment, the adopted datasets and metrics, as well as on the used strategy and the experimental results; some concluding remarks and future work are given in the last Section 6.

2 Background and Related Work

A large number of *credit scoring* classification techniques have been proposed in literature [11], as well as many studies aimed to compare their performance on the basis of several datasets, such as in [8], where a large scale benchmark of 41 classification methods has been performed across eight *credit scoring* datasets.

The problem of how to choose the best classification approach and how to tune optimally its parameters was instead faced in [12]; in the same work are reported some useful observations about the canonical metrics of performance used in this field [13].

2.1 Credit Scoring Models

Most of the statistical and data mining techniques at the state of the art can be used in order to build *credit scoring* models [14,15], e.g., linear discriminant models [16], logistic regression models [3], neural network models [17,18], genetic programming models [19,20], k-nearest neighbor models [21], and decision tree models [22,23].

These techniques can also be combined in order to create hybrid approaches of *credit scoring*, as that proposed in [24,25], based on a two-stage hybrid modeling procedure with artificial neural networks and multivariate adaptive regression splines, or that presented in [26], based on neural networks and clustering methods.

2.2 Imbalanced Class Distribution

A complicating factor in the *credit scoring* process is the imbalanced class distribution of data [27,7], caused by the fact that the default cases are much smaller than the non-default ones. Such distribution of data reduces the performance of the classification techniques, as reported in the study made in [9].

The misclassification costs during the processes of scorecard construction and those of the classification were studied in [28], where it is also proposed to preprocess the dataset of training through an *over-sampling* or a *under-sampling* of the classes. Its effect on the performance has been deeply studied in [29,30].

2.3 Cold Start

The *cold start* problem [31,32] arises when there are not enough information to train a reliable model about a domain [33,34,35].

In the credit scoring context, this happens when there are not many instances related to the credit-worthy and non-credit-worthy customers [36,37]. Considering that, during the model definition, the proposed approach does not exploit the data about the defaulting loans, it is able to reduce/overcome the aforementioned issue.

2.4 Random Forests

Random Forests represent an ensemble learning method for classification and regression that is based on the construction of a number of randomized decision trees during the training phase and it infers conclusions by averaging the results.

Since its formalization [38], it represents one of the most common techniques for data analysis, thanks to its better performance w.r.t. the other state-of-the-art techniques. This technique allows us to face a wide range of prediction problems, without performing any complex configuration, since it only requires the tuning of two parameters: the number of trees and the number of attributes used to grow each tree.

2.5 Shannon Entropy

The Shannon entropy, formalized by *Claude E. Shannon* in [39], is one of the most important metrics used in information theory. It reports the uncertainty associated with a random variable, allowing us to evaluate the average minimum number of bits needed to encode a string of symbols, based on their frequency.

More formally, given a set of values $v \in V$, the entropy $H(V)$ is defined as shown in the Equation 1, where $P(v)$ is the probability that the element v is present in the set V .

$$H(V) = - \sum_{v \in V} P(v) \log_2[P(v)] \quad (1)$$

For instance, if we have a symbol set $V = \{v_1, v_2, v_3, v_4, v_5\}$ where the symbol occurrences in terms of frequency are $v_1 = 0.5, v_2 = 0.2, v_3 = 0.1, v_4 = 0.1, v_5 = 0.1$, the entropy $H(V)$ (i.e., the average minimum number of bits needed to represent a symbol) is given by the Equation 2. Rounding up the result, we need 2 bits/per symbol. So, to represent a sequence of five characters optimally, we need 10 bits.

$$H(V) = - \sum_{v \in V} P(v) \log_2[P(v)] = 1.9 \quad (2)$$

In the context of the classification methods, the use of entropy-based metrics is typically restricted to the *feature selection* [40,41,42], the process where a subset of relevant features (variables, predictors) is selected and used for the definition of the classification model.

In this work, we instead use this metric to detect *anomalous* values in the features of a new instance, where *anomalous* stands for values different from those in the history of the non-default cases.

3 Notation and Problem Definition

This section introduces some notational conventions used in this paper and defines the faced problem.

3.1 Notation

Given a set of classified instances $T = \{t_1, t_2, \dots, t_N\}$, and a set of features $F = \{f_1, f_2, \dots, f_M\}$ that compose each t , we denote as $T_+ \subseteq T$ the subset of non-default instances, and as $T_- \subseteq T$ the subset of default ones.

We also denote as $\hat{T} = \{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_U\}$ a set of unclassified instances and as $E = \{e_1, e_2, \dots, e_U\}$ these instances after the classification process, thus $|\hat{T}| = |E|$.

Each instance can belong only to one class $c \in C$, where $C = \{accepted, rejected\}$.

3.2 Problem Definition

On the basis of the Λ and γ information (explained in Section 4.1), calculated before and after we added to the set T_+ the unclassified instances in the set \hat{T} (one by one), we classify each instance $\hat{t} \in \hat{T}$ as *accepted* or *rejected*.

Given a function $eval(\hat{t}, \lambda, \gamma)$ created to evaluate the correctness of the \hat{t} classification made by exploiting the λ and γ information, which returns a boolean value σ ($0 = \text{misclassification}$, $1 = \text{correct classification}$), we formalize our objective as the maximization of the results sum, as shown in Equation 3.

$$\max_{0 \leq \sigma \leq |\hat{T}|} \sigma = \sum_{u=1}^{|\hat{T}|} eval(\hat{t}_u, \lambda, \gamma) \quad (3)$$

4 Our Approach

The implementation of our approach is carried out through the following three steps:

1. **Local Maximum Entropy by Features** : calculation of the *local maximum entropy by features* Λ , aimed to obtain information about the maximum level of entropy assumed by each feature in the set T_+ ;
2. **Global Maximum Entropy** : calculation of the global maximum entropy γ , a meta-feature defined on the basis of the integral of the area under curve of the *maximum entropy by features* Λ ;
3. **Difference in Maximum Entropy** : formalization of the *Difference in Maximum Entropy (DME)* algorithm, able to classify a new instance as *accepted* or *rejected*, on the basis of the Λ and γ information.

In the following, we provide a detailed description of each of these steps.

4.1 Local Maximum Entropy by Features

Denoting as $H(f)$ the entropy measured in the values assumed by a feature $f \in F$ in the set T_+ , we define the set Λ as shown in Equation 4. It contains the maximum entropy achieved by each $f \in F$, so we have that $|\Lambda| = |F|$. We use this information during the evaluation process explained in Section 4.3.

$$\Lambda = \{\lambda_1 = \max(H(f_1)), \lambda_2 = \max(H(f_2)), \dots, \lambda_M = \max(H(f_M))\} \quad (4)$$

4.2 Global Maximum Entropy

We denote as *global maximum entropy* γ the integral of the area under curve of the *maximum entropy by features* Λ (previously defined in Section 4.1), as shown in Fig. 1.

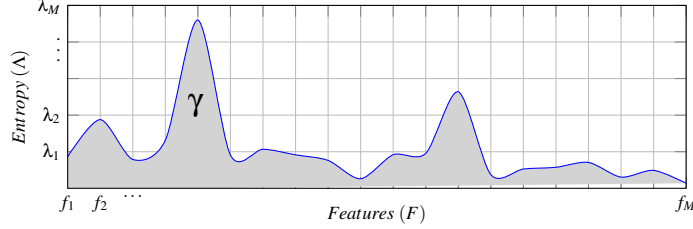


Fig. 1: Global Maximum Entropy γ

More formally, the value of γ is calculated by using the trapezium rule, as shown in Equation 5.

It is a meta-feature that gives us information about the maximum entropy achieved by all features in T_+ , before and after we added to it a unevaluated instance. We use this information during the evaluation process (Section 4.3), jointly with that given by Λ .

$$\gamma = \int_{\lambda_1}^{\lambda_M} f(x) dx \approx \frac{\Delta x}{2} \sum_{k=1}^{|\Lambda|} (f(x_{k+1}) + f(x_k)) \quad \text{with } \Delta x = \frac{(\lambda_M - \lambda_1)}{|\Lambda|} \quad (5)$$

4.3 Difference in Maximum Entropy

The *Difference in Maximum Entropy* (DME) Algorithm 1 is aimed to evaluate and classify a set of unevaluated instances.

It takes as input a set T_+ of non-default instances occurred in the past and a set \hat{T} of unevaluated instances, returning as output a set E containing all instances in \hat{T} , classified as *accepted* or *rejected* on the basis of the Λ and γ information.

In *step 2* we calculate the Λ_a value by using the non-default instances in T_+ , as described in Section 4.1, while in *step 3* we obtain the global maximum entropy γ (Section 4.2). The *steps* from 4 to 26, process all the instances $\hat{t} \in \hat{T}$.

After the calculation of the Λ_b and γ_b values (*steps 5* and *6*), performed by adding the current instance \hat{t} to the non-default instances set T_+ , in the *steps* from 7 to 13, we compare each $\lambda_a \in \Lambda_a$ with the corresponding feature $\lambda_b \in \Lambda_b$ (*steps* from 8 to 12), counting how many times the value of λ_b is greater than that of λ_a , increasing the value of b (*step 9*) when this happens, or that of a otherwise (*step 11*); in the *steps* from 14 to 18 we perform the same operation, but by taking into account the global maximum entropy γ .

At the end of the previous sub-processes, in the *steps* from 19 to 23 we classify the current instance as *accepted* or *rejected*, on the basis of the a and b values, then we set them to zero (*steps 24* and *25*). The resulting set E is returned at the end of the entire process (*step 27*).

Algorithm 1 *Difference in Maximum Entropy (DME)*

Input: T_+ =Set of non-default instances, \hat{T} =Set of instances to evaluate
Output: E =Set of classified instances

```

1: procedure INSTANSESEVALUATION( $T_+, \hat{T}$ )
2:    $\Lambda_a$ =getLocalMaxEntropy( $T_+$ )
3:    $\gamma_a$ =getGlobalMaxEntropy( $\Lambda_a$ )
4:   for each  $\hat{i}$  in  $\hat{T}$  do
5:      $\Lambda_b$ =getLocalMaxEntropy( $T_+ + \hat{i}$ )
6:      $\gamma_b$ =getGlobalMaxEntropy( $\Lambda_b$ )
7:     for each  $\lambda$  in  $\Lambda$  do
8:       if  $\lambda_b > \lambda_a$  then
9:          $b++$ 
10:       else
11:          $a++$ 
12:       end if
13:     end for
14:     if  $\gamma_b > \gamma_a$  then
15:        $b++$ 
16:     else
17:        $a++$ 
18:     end if
19:     if  $b > a$  then
20:        $E \leftarrow (\hat{i}, rejected)$ 
21:     else
22:        $E \leftarrow (\hat{i}, accepted)$ 
23:     end if
24:    $a = 0;$ 
25:    $b = 0;$ 
26: end for
27: return  $E$ 
28: end procedure

```

5 Experiments

This section describes the experimental environment, the used datasets and metrics, the adopted strategy, and the results of the performed experiments.

5.1 Experimental Setup

The proposed approach was developed in Java, while the implementation of the state-of-the-art approach used to evaluate its performance was made in R^1 , using the *random-Forest* and *ROCR* packages.

The experiments have been performed by using two real-world datasets characterized by a strong unbalanced distribution of data. For reasons of reproducibility of the *RF* experiments, the *R* function *set.seed()* has been used in order to fix the seed of the random number generator. The *RF* parameters have been tuned by searching those that maximize the performance.

It should be further added that we verified the existence of a statistical difference between the results, by using the independent-samples *two-tailed Student's t-tests* ($p < 0.05$).

¹ <https://www.r-project.org/>

5.2 DataSets

The two real-world datasets used in the experiments (i.e., *Default of Credit Card Clients dataset* and *German Credit* datasets, both available at the *UCI Repository of Machine Learning Databases*²) represent two benchmarks in this research field. In the following we provide a short description of their characteristics:

Default of Credit Card Clients (DC). It contains 30,000 instances: 23,364 of them are credit-worthy applicants (77.88%) and 6,636 are not credit-worthy (22.12%). Each instance contains 23 attributes and a binary class variable (*accepted* or *rejected*).

German Credit (GC). It contains 1,000 instances: 700 of them are credit-worthy applicants (70.00%) and 300 are not credit-worthy (30.00%). Each instance contains 21 attributes and a binary class variable (*accepted* or *rejected*).

5.3 Metrics

This section presents the metrics used in the experiments.

Accuracy. The *Accuracy* metric reports the number of instances correctly classified, compared to the total number of them. More formally, given a set of instances X to be classified, it is calculated as shown in Equation 6, where $|X|$ stands for the total number of instances, and $|X^{(+) }|$ for the number of those correctly classified.

$$Accuracy(X) = \frac{|X^{(+)}|}{|X|} \quad (6)$$

F-measure. The *F-measure* is the weighted average of the *precision* and *recall* metrics. It is a largely used metric in the statistical analysis of binary classification and gives us a value in a range $[0, 1]$, where 0 represents the worst value and 1 the best one. More formally, given two sets X and Y , where X denotes the set of performed classifications of instances, and Y the set that contains the actual classifications of them, this metric is defined as shown in Equation 7.

$$F\text{-measure}(X, Y) = 2 \cdot \frac{(\text{precision}(X, Y) \cdot \text{recall}(X, Y))}{(\text{precision}(X, Y) + \text{recall}(X, Y))} \quad (7)$$

with

$$\text{precision}(X, Y) = \frac{|Y \cap X|}{|X|}, \quad \text{recall}(X, Y) = \frac{|Y \cap X|}{|Y|}$$

² <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/>

AUC. The *Area Under the Receiver Operating Characteristic* curve (*AUC*) is a performance measure used to evaluate the effectiveness of a classification model [43,44]. Its result is in a range $[0, 1]$, where 1 indicates the best performance. More formally, according with the notation of Section 3, given the subset of non-default instances T_+ and the subset of default ones T_- , the formalization of the *AUC* metric is reported in the Equation 8, where Θ indicates all possible comparisons between the instances of the two subsets T_+ and T_- . It should be noted that the result is obtained by averaging over these comparisons.

$$\Theta(t_+, t_-) = \begin{cases} 1, & \text{if } t_+ > t_- \\ 0.5, & \text{if } t_+ = t_- \\ 0, & \text{if } t_+ < t_- \end{cases} \quad AUC = \frac{1}{|T_+| \cdot |T_-|} \sum_1^{|T_+|} \sum_1^{|T_-|} \Theta(t_+, t_-) \quad (8)$$

5.4 Strategy

The experiments have been performed by using the *k-fold cross-validation* criterion, with $k=10$. Each dataset is randomly shuffled and then divided in k subsets; each k subset is used as test set, while the other $k-1$ subsets are used as training set, considering as result the average of all results. This approach allows us to reduce the impact of data dependency and improves the reliability of the results.

5.5 Experimental Results

As shown in Fig. 2 and Fig. 3, the performance of our *DME* approach is very similar to the *RF* one, both in terms of *Accuracy* and in terms of *F-measure*, where we achieve better performance than *RF* with the *DC* dataset.

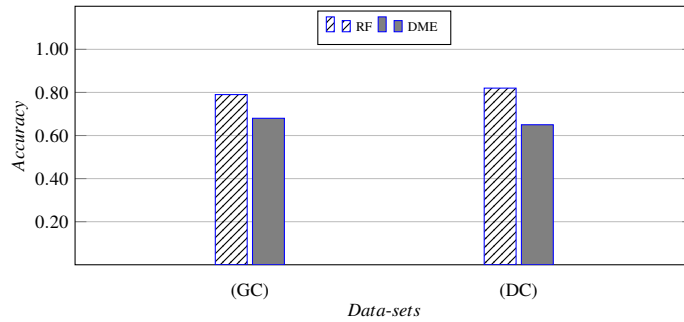
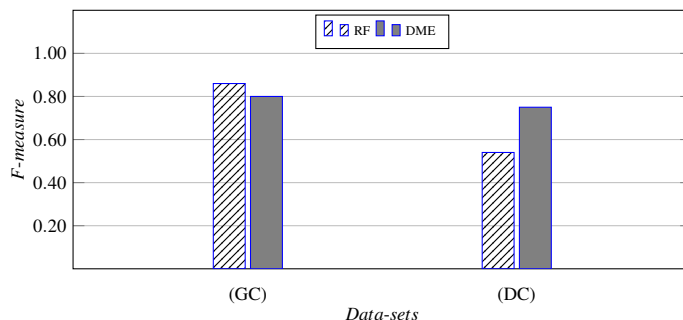


Fig. 2: Accuracy Performance

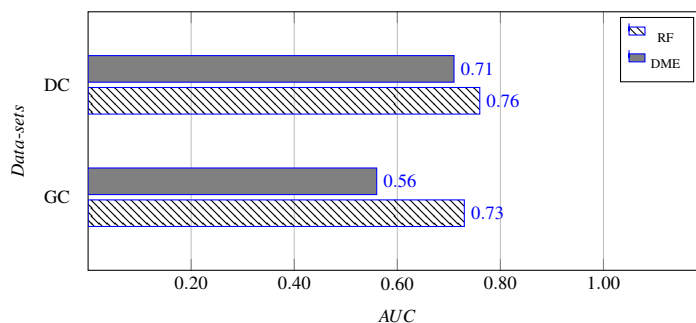
By examining the obtained results, the first observation that rises is related to the fact that our approach gets the same performance of *RF*, despite it operates in a proactive manner (i.e., without using default cases during the training process).

Fig. 3: *F-measure Performance*

Another observation arises about the *F-measure* results, which show how the effectiveness of our approach increases with the number of non-default instances used in the training process (*DC* dataset). This does not happen with *RF*, although it uses both default and non-default instances, during the model training.

We can observe interesting results also in terms of *AUC*: this metric evaluates the predictive capability of a classification model, and the results in Fig. 4 show that our performance is similar to those of *RF*, although we did not train our model with both classes of instances.

It should be noted that, as introduced in Section 1, the capability of the *DME* approach to operate proactively allows us to reduce/overcome the cold-start problem.

Fig. 4: *AUC Performance*

6 Conclusions and Future Work

The *credit scoring* techniques cover a crucial role in many financial contexts (i.e., personal loans, insurance policies, etc.), since they are used by financial operators in order to evaluate the potential risks of lending, allowing them to reduce the losses due to default.

This paper proposes a novel approach of *credit scoring* that exploits an entropy-based criterion to classify a new instance as *accepted* or *rejected*.

Considering that it does not need to be trained with the past default instances, it is able to operate in a proactive manner, also reducing/overcoming the cold-start and the data imbalance problems that reduce the effectiveness of the canonical machine learning approaches.

The experimental results presented in Section 5.5 show two important aspects related to our approach: on the one hand, it performs similarly to one of the best performing approaches in the state of the art (i.e., *RF*), by operating proactively; on the other hand, it is able to outperform *RF*, when in the training process a large number of non-default instances are involved, differently from *RF*, where the performance does not improve further.

A possible follow up of this paper could be a new series of experiments aimed at improving the non-proactive state-of-the-art approaches, by adding the information related to the default cases, as well as the evaluation of the proposed approach in heterogeneous scenarios, which involve different types of financial data, such as those generated by an electronic commerce environment.

Acknowledgments. This work is partially funded by Regione Sardegna under project NOMAD (Next generation Open Mobile Apps Development), through PIA - Pacchetti Integrati di Agevolazione “Industria Artigianato e Servizi” (annualità 2013).

References

1. Henley, W., et al.: Construction of a k-nearest-neighbour credit-scoring system. *IMA Journal of Management Mathematics* **8**(4) (1997) 305–321
2. Mester, L.J., et al.: Whats the point of credit scoring? *Business review* **3** (1997) 3–16
3. Henley, W.E.: Statistical aspects of credit scoring. PhD thesis, Open University (1994)
4. Fensterstock, A.: Credit scoring and the next step. *Business Credit* **107**(3) (2005) 46–49
5. Brill, J.: The importance of credit scoring models in improving cash flow and collections. *Business Credit* **100**(1) (1998) 16–17
6. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter* **6**(1) (2004) 20–29
7. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intell. Data Anal.* **6**(5) (2002) 429–449
8. Lessmann, S., Baesens, B., Seow, H., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* **247**(1) (2015) 124–136
9. Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.* **39**(3) (2012) 3446–3453
10. Bhattacharyya, S., Jha, S., Tharakunnel, K.K., Westland, J.C.: Data mining for credit card fraud: A comparative study. *Decision Support Systems* **50**(3) (2011) 602–613
11. Doumpos, M., Zopounidis, C.: Credit scoring. In: *Multicriteria Analysis in Finance*. Springer (2014) 43–59
12. Ali, S., Smith, K.A.: On learning algorithm selection for classification. *Appl. Soft Comput.* **6**(2) (2006) 119–138
13. Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning* **77**(1) (2009) 103–123

14. Chen, S.Y., Liu, X.: The contribution of data mining to information science. *Journal of Information Science* **30**(6) (2004) 550–558
15. Alborzi, M., Khanbabaie, M.: Using data mining and neural networks techniques to propose a new hybrid customer behaviour analysis and credit scoring model in banking services based on a developed RFM analysis method. *IJBIS* **23**(1) (2016) 1–22
16. Reichert, A.K., Cho, C.C., Wagner, G.M.: An examination of the conceptual issues involved in developing credit-scoring models. *Journal of Business & Economic Statistics* **1**(2) (1983) 101–114
17. Desai, V.S., Crook, J.N., Overstreet, G.A.: A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research* **95**(1) (1996) 24–37
18. Blanco-Oliver, A., Pino-Mejías, R., Lara-Rubio, J., Rayo, S.: Credit scoring models for the microfinance industry using neural networks: Evidence from peru. *Expert Syst. Appl.* **40**(1) (2013) 356–364
19. Ong, C.S., Huang, J.J., Tzeng, G.H.: Building credit scoring models using genetic programming. *Expert Systems with Applications* **29**(1) (2005) 41–47
20. Chi, B., Hsu, C.: A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. *Expert Syst. Appl.* **39**(3) (2012) 2650–2661
21. Henley, W., Hand, D.J.: A k-nearest-neighbour classifier for assessing consumer credit risk. *The Statistician* (1996) 77–95
22. Davis, R., Edelman, D., Gamberman, A.: Machine-learning algorithms for credit-card applications. *IMA Journal of Management Mathematics* **4**(1) (1992) 43–51
23. Wang, G., Ma, J., Huang, L., Xu, K.: Two credit scoring models based on dual strategy ensemble trees. *Knowl.-Based Syst.* **26** (2012) 61–68
24. Lee, T.S., Chen, I.F.: A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications* **28**(4) (2005) 743–752
25. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. *Expert Syst. Appl.* **38**(1) (2011) 223–230
26. Hsieh, N.C.: Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications* **28**(4) (2005) 655–665
27. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9) (2009) 1263–1284
28. Vinciotti, V., Hand, D.J.: Scorecard construction with unbalanced class sizes. *Journal of Iranian Statistical Society* **2**(2) (2003) 189–205
29. Marqués, A.I., García, V., Sánchez, J.S.: On the suitability of resampling techniques for the class imbalance problem in credit scoring. *JORS* **64**(7) (2013) 1060–1070
30. Crone, S.F., Finlay, S.: Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting* **28**(1) (2012) 224–238
31. Zhu, J., Wang, H., Yao, T., Tsou, B.K.: Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In Scott, D., Uszkoreit, H., eds.: *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK.* (2008) 1137–1144
32. Donmez, P., Carbonell, J.G., Bennett, P.N.: Dual strategy active learning. In: *ECML. Volume 4701 of Lecture Notes in Computer Science.*, Springer (2007) 116–127
33. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. *Expert Syst. Appl.* **41**(4) (2014) 2065–2073
34. Son, L.H.: Dealing with the new user cold-start problem in recommender systems: A comparative review. *Inf. Syst.* **58** (2016) 87–104

35. Fernández-Tobías, I., Tomeo, P., Cantador, I., Noia, T.D., Sciascio, E.D.: Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In Sen, S., Geyer, W., Freyne, J., Castells, P., eds.: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, ACM (2016) 119–122
36. Attenberg, J., Provost, F.J.: Inactive learning?: difficulties employing active learning in practice. *SIGKDD Explorations* **12**(2) (2010) 36–41
37. Thanuja, V., Venkateswarlu, B., Anjaneyulu, G.: Applications of data mining in customer relationship management. *Journal of Computer and Mathematical Sciences* Vol **2**(3) (2011) 399–580
38. Breiman, L.: Random forests. *Machine Learning* **45**(1) (2001) 5–32
39. Shannon, C.E.: A mathematical theory of communication. *Mobile Computing and Communications Review* **5**(1) (2001) 3–55
40. Dash, M., Liu, H.: Feature selection for classification. *Intell. Data Anal.* **1**(1-4) (1997) 131–156
41. Kwak, N., Choi, C.: Input feature selection for classification problems. *IEEE Trans. Neural Networks* **13**(1) (2002) 143–159
42. Jiang, F., Sui, Y., Zhou, L.: A relative decision entropy-based feature selection approach. *Pattern Recognition* **48**(7) (2015) 2151–2163
43. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. (2011)
44. Faraggi, D., Reiser, B.: Estimation of the area under the roc curve. *Statistics in medicine* **21**(20) (2002) 3093–3106