A Combined Entropy-based Approach for a Proactive Credit Scoring

Salvatore Carta, Anselmo Ferreira, Diego Reforgiato Recupero, Marco Saia, Roberto Saia

Department of Mathematics and Computer Science
University of Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy

Abstract

Lenders, such as credit card companies and banks, use credit scores to evaluate the potential risk posed by lending money to consumers and, therefore, mitigating losses due to bad debt. Within the financial technology domain, an ideal approach should be able to operate proactively, without the need of knowing the behavior of non-reliable users. Actually, this does not happen because the most used techniques need to train their models with both reliable and non-reliable data in order to classify new samples. Such a scenario might be affected by the cold-start problem in datasets, where there is a scarcity or total absence of non-reliable examples, which is further worsened by the potential unbalanced distribution of the data that reduces the classification performances. In this paper, we overcome the aforementioned issues by proposing a proactive approach, composed of a combined entropy-based method that is trained considering only reliable cases and the sample under investigation. Experiments done in different real-world datasets show competitive performances with several state-of-art approaches that use the entire dataset of reliable and unreliable cases.

Keywords: FinTech, Trust Management, Business Intelligence, Credit Scoring, Data Mining, Entropy

Email address: {salvatore, anselmo.ferreira, diego.reforgiato, roberto.saia }@unica.it, m.saia@studenti.unica.it (Salvatore Carta, Anselmo Ferreira, Diego Reforgiato Recupero, Marco Saia, Roberto Saia)

1. Introduction

The main task of a *Credit Scoring* system is the evaluation of new loan applications (from now on named *instances*) in terms of their potential reliability. Its goal is to lead the financial operators toward a decision about accepting or not a new credit, on the basis of a reliability score assigned by the Credit Scoring system [1]. In a nutshell, the Credit Scoring system is a statistical approach able to evaluate the probability that a new instance is considered reliable (non-default) or unreliable (default), by exploiting a model defined on the basis of previous instances [2, 3]. Banks and credit card companies use credit scores to determine who qualifies for a loan, at what interest rate, and at what credit limits. Therefore, Credit Scoring systems reduce losses due to default cases [4], and, for this reason, they represent a crucial instrument. Although similar technical issues are shared, Credit Scoring is different from Fraud detection, which consists of a set of activities undertaken to prevent money or property from being obtained through false pretenses.

Thanks to their capability to analyze all the components that contribute to determine default cases [5], Credit Scoring techniques can also be considered a powerful instrument for risk assessment and real-time monitoring [6].

Moreover, lenders may also use credit scores to determine which customers are likely to bring in the most revenue. However, as usually happens with other similar contexts (*e.g.*, Fraud Detection [7]), the main problem that limits the effectiveness of Credit Scoring classification techniques is represented by the unbalanced distribution of data [8]. This happens because the default cases available for training the evaluation model are fewer than the non-default ones, hampering the performances of machine learning approaches applied to Credit Scoring [9]. To note that the unbalanced distribution of data is one of the problems that enables the cold start problem. As such, approaches for balancing data mitigate the cold start problem as well.

To overcome such an issue, in this paper we evaluate the instances in terms of their features entropy, defining a metric able to measure their level of reliability considering only non-default cases and the instance under investigation. More formally, we evaluate the reliability of a new instance in terms of comparing the Shannon Entropy (from now on referred simply as *entropy*) measured within a set of previous non-default instances before and after adding the instance under investigation. As the entropy measures the uncertainty of a random variable, a larger entropy in the set including the sample investigated indicates that it contains similar data in its features, which increases the level of equiprobability, and then we tend to classify it as reliable. Otherwise, it contains different data and we consider the instance as unreliable. Such a process allows us operating proactively, overcoming the issue related to the unbalanced distribution of the data and, at the same time, mitigating the cold-start problem (*i.e.*, the scarcity or total absence of default examples).

We report comparisons between our approach and Random Forests, which are considered state-of-the-art approaches for credit scoring tasks [10, 11, 12]. For that we used two real-world datasets, characterized by different distribution of data (unbalanced and slightly unbalanced). Experiments results show that, although our approach is trained on reliable cases only, it has similar performances to the Random Forests.

Therefore, the main scientific contributions given by this paper are listed below:

(i) Calculation of the *Local Entropy* in the process of credit scoring, a process aimed to measure the entropy achieved by each feature in the previous non-default instances, in order to evaluate the entropy variations in terms of single features of an instance.

(ii) Calculation of the *Global Entropy* in the process of credit scoring, a meta-feature obtained by calculating the integral of the area under curve given by the local entropies, which allows us evaluating the entropy variations in terms of all features of an instance.

(iii) Definition of the *Entropy Difference Approach*, an algorithm able to classify the new instances as reliable or unreliable by exploiting both the *Local Entropy* and *Global Entropy* information.

This paper is based on a previous work [13], which has been completely revised, rewritten, improved and extended with the following novel contributions:

1. We updated our proposed approach by defining a threshold of differences in the process of instance classification, aiming at optimizing the performance on the basis of the specific operative context, differently from our previous formalization [13] based on comparing two counters.

2. A feature selection step is now done in our proposed approach, in order to select instance features based on a twofold criterion (*i.e.*, basic and mutual entropy). Additionally, experiment comparisons between the performance achieved by our approach before and after we performed the proposed feature selection process are reported, to better highlight the benefits of such a pre-processing step.

3. A complexity analysis is added by evaluating the asymptotic time complexity of the proposed algorithm, in order to determinate its impact in some particular contexts such as real-time Credit Scoring system, a process not done in our previous work [13].

4. One more dataset, which is more suitable for the scenario taken into account (*i.e.*, the Australian Credit Approval dataset), is added to the experiments, allowing us to better evaluate the performance of our approach in two different data configurations (highly unbalanced and slightly unbalanced).

5. We added a new metric of evaluation (*i.e.*, Sensitivity) in the experiments, which allows us to have a detailed overview of the proposed approach performance.

6. We added experiments results of the parameter tuning process aimed at

finding the best threshold of the proposed algorithm, which was not reported in our previous work [13].

7. We added three more baselines based on improved Naive-Bayes classifiers as competitors.
8. We performed one experiment of varying the number of default (minority class) samples available to the classifiers, better highlighting the benefits of the proposed approach in a real world credit scoring scenario.

The remainder of the paper is organized as follows. Section 2 discusses the background and related works of credit scoring. Section 3 describes the implementation of the proposed approach. Section 4 provides details on the experimental environment, the adopted datasets and metrics, as well as on the implementation of the proposed approach and the competitors. Section 5 shows the experimental results and, finally, some concluding remarks and future work are given in Section 6.

2. Related Works

The research related to the Credit Scoring has grown quite significantly in recent years, in coincidence with the exponential increase of consumer credit [14]. The literature proposes a large number of Credit Scoring techniques [15, 16, 17] to maximize Equation 1, along with several studies focused on comparing their performance in several real-world datasets. We discuss some of such solutions in the remaining of this section.

The work in [18] used the Wavelet transform and three metrics to perform credit scoring. Similarly, the approach in [19] moved the credit scoring from the canonical time domain to the frequency one, by comparing differences of magnitudes after Fourier Transform conversion of time-series data. An interesting approach was proposed in [20], which presents a comparison of non-square matrix determinants identify the reliability of users data to allow money loan. The work in [21] used a score based on outlier parameters for each transaction, together with an isolation forest classifier to detect unreliable users. Kolmogorov-Smirnov statistics were used in [22] to cluster unreliable and reliable users. Authors of [23] used data preprocessing and a Random Forest optimized through a grid search step. A three-way decisions approach with probabilistic rough sets is proposed in [24]. In [25], a deep learning Convolutional Neural Network approach is used for the first time for credit scoring, which is applied to features that are pre-processed with the Relief feature selection technique and converted into grayscale images. An application of kernel-free fuzzy quadratic surface Support Vector Machines is proposed in [26], and an interesting comparison of different neural networks, such as Multilayer Perceptrons and Convolutional Neural Networks for Credit Scoring is done in [27]. An extensive work in this sense was done in [10], where a large scale benchmark of forty-one methods for the instance classification has been performed on eight Credit Scoring datasets. Another type of problem, related to the optimization of the parameters involved in these approaches was instead tackled in [28], which also reports a discussion about the canonical metrics used to measure the performance [29].

Machine learning techniques can also be combined in order to build hybrid approaches of Credit Scoring as, for instance, those presented in [30, 31], which exploit a two-stage hybrid model with artificial neural networks and a multivariate adaptive regression splines model, or that described in [32], which instead exploits neural networks with k-mean clustering method. Another kind of classifiers combination, commonly known as *ensembles*, has also been extensively studied in the literature. The work in [33] used several classifiers, including SVMs and logistic regression, in order to validate a feature selection approach, called *group penalty function*, which penalizes the use of variables from the same source of information in the final features. In [34], a multi-step data processing operation that includes normalization and dimensionality reduction, allied with an ensemble of five classifiers optimized by a Bayesian algorithm, are used in the pipeline. The work in [35] ensembles five classifiers (logistic regression, support vector machine, neural network, gradient boosting decision tree and random forest) using a genetic algorithm and fuzzy assignment. In [36], a set of classifiers are joined in an ensemble according to their soft probabilities. In [37], an ensemble is used with a feature selection step based on feature clustering, and the final result is a weighted voting approach.

Other works are closely related and can be integrated to Credit Scoring application. For example, in *user profiling*, users can be considered good and bad borrowers, not only according to core credit information, but also their behavior in social networks. In this sense, the work in [38] used a Naive-Bayes based classifier in both features: hard (credit information) and soft (friendship and group information). Linguistic-based features are coupled with machine learning classifiers in [39] to detect a person's behavior. Finally, the work in [40] used deep learning through Long Short Term Memory networks on texts to define the personality of a person.

Notwithstanding, several issues and limitations are still considered open problems in Credit Scoring tasks. We discuss all of them in the following:

1. **Data Scarcity Problem**: this issue refers to the lack of data to validate machine learning models [41]. This happens mainly due to the policies and constraints adopted by researchers working in this field, which do not allow them releasing information about their business activities for privacy, competition, or legal issues.

2. **Non-adaptability Problem** this problem concerns the inability of the Credit Scoring models to correctly classify the new instances, especially when their features generate different patterns w.r.t the patterns used to define the evaluation model. All the Credit Scoring approaches are affected by this problem that leads toward misclassification, due to their inability to identify new patterns in the instances under analysis.

3. **Data Heterogeneity Problem**: the pattern recognition process used to detect some specific patterns on the basis of a model previously defined represents a very important branch of the machine learning, since it can

be used to solve a large number of real-world problems [42]. However, it should be noted how the effectiveness of these processes can be reduced by the heterogeneity of the involved data. Such a problem, also known in literature as *instance identification* or *naming problem*, is due to the fact that same data are often represented in a different way in different datasets [43].

4. **Cold-start Problem**: such an issue arises when the set of data used to train an evaluation model does not contain enough information about the domain taken into account, making it impossible to define a reliable model [44, 45, 46]. In other words, this happens when the training data are not representative of all the involved classes of information [47, 48], which in the application discussed herein are represented by the default and non-default cases. More formally, within the credit scoring domain, the cold start problem consists of the following three cases: (i) *New community.* When a catalogue of financial indicators exist but almost no users are present and the lack of user interaction makes it very hard to provide reliable suggestions. (ii) *New financial feature.* A new financial feature is added to the system but there are no interactions (financial features applicable to a given user) present. (iii) *New user.* A new user registers but he/she has not provided any interaction yet, therefore it is not possible to provide personalized analysis.

5. **Data Unbalance Problem**: without underestimating the other problems, we can state that the main complicating factor in a Credit Scoring process is the imbalanced class distribution of data [49, 9], caused by the fact that the default cases are much smaller than the non-default ones. This means that the information available to train an evaluation model is typically composed of a large number of legitimate cases and a small number of fraudulent ones, a data configuration that reduces the effectiveness of the most common classification approaches [9, 11]. A common solution adopted in order to face this problem is the artificial balance of data [50].

It consists of an over-sampling or under-sampling operation. In the first case the balance is obtained by duplicating some of the instances that occur the least (usually, the default ones), while in the second case it is obtained by removing some of the instances that occur the most (usually, the non-default ones). An analysis of the advantages and disadvantages related to this preprocessing phase has been presented in [51, 52].

Some works have focused on the problem of imbalanced learning in datasets. In [53], the authors presented a technique that clones the minority class instances according to the similarity between them and the minority class mode. The work in [54] proposed cost-sensitive Bayesian network classifiers, which incorporate an instance weighting method giving different classification errors to different classes. Authors in [55] proposed undersampling and oversampling approaches based on a novel class imbalance metric, which splits the imbalance problem into multiple balanced subproblems. Then, weak classifiers trained in a bagging manner are used in a boosting fashion. The approach proposed in [56] capture the covariance structure of the minority class in order to generate synthetic samples with Mahalanobis Distance-based Over-sampling and Generalized Singular Value Decomposition. The research performed in [57] studied potential bias characteristics of imbalanced crowdsourcing labeled datasets. Then, the authors proposed a novel consensus algorithm based on weighted majority voting of four classifiers. Such algorithm uses the frequency of minority class to obtain a bias rate, assigning weights to the majority and minority classes. The authors of [58] enhanced a multi-class classifier based on fuzzy rough sets. Firstly, they proposr an adaptive weight setting for the binary classifiers involved, addressing the varying characteristics of sub-problems. Then, a new dynamic aggregation method combines the predictions of binary classifiers with a global class affinity method before making a final decision. Finally, authors in [59] evolved one-vs-one schemes for multi-class imbalance classification problems, by applying binary ensemble learning approaches with an aggregation approach.

However, differently from all of these previous approaches, our method doesn't need any samples from the minority class in the proposed pipeline, a problem that can happen specially when the cold-start problem arises (*i.e.*, there is no default cases in the dataset). Our approach faces these problems by training its evaluation model using only one class of data (the non-default cases, or the majority class), comparing entropy-based metrics behavior of non-evaluated samples before and after they are added to a set of previous non-default samples. Therefore, our proposed approach represents a side effect of adopting a proactive methodology by being aware of limitations of the environment. We discuss further details of our proposed approach in the next section.

3. Proposed Approach

Before we discuss our solution for the credit score in more details, let us define the problem of Credit Scoring more formally. Given a set of classified instances $T = \{t_1, t_2, \dots, t_K\}$ and a set of features $F = \{f_1, f_2, \dots, f_M\}$ that compose each $t \in T$, we denote as $T_+ = \{t_1, t_2, \dots, t_N\}$ the subset of non-default instances (then $T_+ \subseteq T$), and as $T_- = \{t_1, t_2, \dots, t_J\}$ the subset of default ones (then $T_- \subseteq T$). 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 $|\hat{T}| = |E|$). It should be observed that an instance can only belong to one class $c \in C$, where $C = \{reliable, unreliable\}$. So, the Credit Score system problem is to define a function $eval(\hat{t}_u)$ which returns the maximum sum of a binary value σ, used to assess the correctness of $\hat{t}_u$ classification (*i.e.*, 0=misclassification, 1=correct classification), or

$$\max_{0 \leq \sigma \leq |\hat{T}|} \sigma = \sum_{u=1}^{|\hat{T}|} eval(\hat{t}_u). \tag{1}$$

Given such concepts, the implementation of our approach has been carried out through the following four steps:

1. **Feature Selection Process**: evaluation of each instance feature in order to evaluate its contribution in the context of the definition of our evaluation model.

2. **Local Entropy Calculation**: calculation of the *local entropy* Λ, which gives information about the level of entropy assumed by each single feature in the set T_+.

3. **Global Entropy Calculation**: calculation of the *global entropy* γ, a meta-information defined by calculating the integral of the area under the Λ curve.

4. **Entropy Difference Approach**: definition of the *Entropy Difference Approach* (EDA) able to classify the new instances on the basis of the Λ and γ information.

A pipeline of the proposed EDA approach is shown in Figure 1. In the first step, the set of previous non-default instances T_+ and the set of instances to be evaluated $\hat{T}$ are preprocessed, performing a *feature selection* task aimed to exclude from the evaluation process the features with a low level of characterization of the instances. This step reduces the computational complexity and returns sets with reduced features T'_+ and $\hat{T}'$. In the next steps, the local entropy is calculated for each feature of the set T'_+, as well as the global entropy of all the features in T'_+. The last step performs the comparison between the local and global entropy previously calculated for the set T'_+, and the same information calculated for adding each element of the set $\hat{T}'$ to T'_+, classifying the non evaluated instances on the basis of the threshold Θ. The result of the entire process is then stored in the set E.

Algorithm 1 describes the general idea of the approach and is composed of two steps. It receives as input the set T_+ of reliable instances, the set $\hat{T}$ of non-evaluated instances and three thresholds: $min1$, and $min2$ from the feature selection approach, and Θ from the proposed EDA approach. The first step calculates the reduced features using basic and mutual Shanon entropies metrics to eliminate features according to thresholds $min1$ and $min2$ (a process further discussed in Section 3.1). The transformed sets T'_+ of reliable instances and $\hat{T}'$ of non-evaluated instances are then the input of the proposed EDA approach

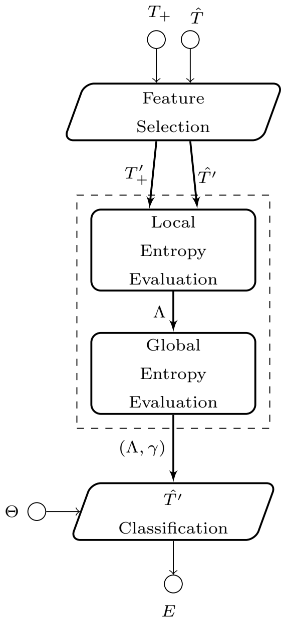


Figure 1: *EDA High-level Architecture*

(Section 3.4), which classifies each $\hat{t}' \in \hat{T}'$ by the threshold θ on comparisons of Local Maximum Entropy (Section 3.2) and global Maximum Entropy (Section 3.3) values calculated before and after adding non evaluated instances $\hat{t}' \in \hat{T}'$ to $\hat{T'_+}$. Then, the set E will return the classification of each non evaluated sample $\hat{t}' \in \hat{T}'$. In the following subsections, we will describe in details all the aforementioned steps.

Algorithm 1 *Proactive Credit Scoring Approach*

```
Input: $T_+$=Set of non-default instances; $\hat{T}$=Set of instances to evaluate; $min_1, min_2$=Basic and mutual entropy thresholds; $\Theta$=EDA Threshold
Output: $E$=Set of classified instances
procedure Proactive_Credit_scoring($T_+$, $\hat{T}$, $min_1$,$min_2$, $\Theta$)
    $T'_+$, $\hat{T}' \leftarrow FeatureSelection(T_+, \hat{T}, min_1, min_2)$        ▷ See Section 3.1
    $E \leftarrow InstancesEvaluation(T'_+, \hat{T}', \Theta)$        ▷ See Sections 3.2, 3.3 and 3.4.
    return $E$
end procedure
```

3.1. Feature Selection

Many studies [60] have discussed how the performance of a Credit Scoring model is strongly influenced by the features used during the process of their definition. This process is known as *Feature Selection* and it can be performed by using different techniques, on the basis of the characteristics of the context taken into account. It means that the choice of the best features to use during the model definition is not based on a unique criterion, but rather it exploits several criteria with the aim to evaluate, as best as possible, the influence of each feature in the process of defining the Credit Scoring model. This represents an important preprocessing step, since it can reduce the complexity of the final model, decreasing the training times and increasing the generalization of the model at the same time. Further, it can also reduce the problem related to the overfitting, a problem that occurs when a statistical model describes random error or noise instead of the underlying relationship, and this frequently happens during the definition of excessively complex models, since many parameters, with respect to the number of training data, are involved.

In the proposed approach, the feature selection is performed by exploiting a *dual entropy-based approach* that evaluates the importance of the features both individually and mutually. For that, we use two metrics, defined as follows.

Basic Shannon Entropy. It measures the uncertainty associated with a random variable by evaluating the average minimum number of bits needed to encode a string of symbols based on their frequency. High values of entropy indicate a high level of uncertainty in the data prediction process and, otherwise, low values of entropy indicate a lower degree of uncertainty in this process. More formally, given a set of values $f \in F$, the entropy $H(F)$ is defined as shown in the Equation 2, where $P(f)$ is the probability that the element f is present in the set F.

$$H(F) = -\textstyle\sum_{f \in F} P(f) log_2[P(f)] \qquad (2)$$

Mutual Shannon Entropy. It measures the amount of information a random variable gives about another one. High mutual information values indicate a large reduction in uncertainty, while low mutual information values indicate a small reduction of uncertainty. A value of zero indicates that the variables are independent. More formally, given two discrete variables X and Y whose joint probability distribution is $P_{XY}(x, y)$, denoting as $\mu(X; Y)$ the mutual information between X and Y, the Mutual Shannon Entropy is calculated as shown in Equation 3 below

$$\mu(X;Y) = \sum_{x,y} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}} \log \frac{P_{XY}}{P_X P_Y}. \tag{3}$$

With these two metrics in mind, we perform the feature selection through the following steps:

1. The basic entropy of each single feature is measured, evaluating its contribution in the instance characterization.
2. The mutual entropy of each feature with respect to the other features is evaluated.
3. Results of the previous two steps are combined, selecting the features to be used within the model definition process.

Such an approach allows us evaluating the contribution of each feature from a dual point of view, by deciding when we can exclude it in order to reduce the computational complexity, an important preprocessing task in case of large datasets.

The feature selection process is detailed in Algorithm 2. It takes as input a set T_+ of previous non-default instances, the set $\hat{T}$ of instances to evaluate and min_1 and min_2 values, which represent the thresholds used to determine when an entropy value must be considered relevant (as previously described). The algorithm returns then two sets of instances, T'_+ and $\hat{T}'$, which contain only the features that had not been removed by the algorithm, in order to use them in the model definition process. In step 2 of the algorithm, we extract the features related to the dataset T_+, processing them in the steps 4-10. Such a process calculates the basic and mutual entropy (steps 5 and 6) in the set of values assumed by each feature in the dataset T_+, removing (steps 8 and 9) from T_+ and $\hat{T}$ the features in T_+ that present a basic entropy above the min_1 value and a mutual entropy below the min_2 value (step 7). At step 12, the sets T'_+ and $\hat{T}'$ with reduced features are returned by the algorithm.

Algorithm 2 *Feature Selection*

```
Input: $T_+$=Set of non-default instances; $\hat{T}$=Set of instances to evaluate; $min_1, min_2$=Basic and mutual entropy thresholds
Output: $T'_+$=Set of non-default instances with selected features; $\hat{T}'$=Set of instances to evaluate with selected features
procedure FeatureSelection($T_+$, $\hat{T}$, $min_1$,$min_2$)
    $F_+ \leftarrow getAllFeatures(T_+)$
    $\hat{F} \leftarrow getAllFeatures(\hat{T})$
    for each $f$ in $F_+$ do
        $be \leftarrow getBasicEntropy(F_+, f)$
        $me \leftarrow getMutualEntropy(F_+, f)$
        if $be > min_1$ AND $me < min_2$ then
            $T'_+ \leftarrow removeFeature(f, F_+)$
            $\hat{T}' \leftarrow removeFeature(f, \hat{F})$
        end if
    end for
    return $T'_+$, $\hat{T}'$
end procedure
```

3.2. Local Maximum Entropy Calculation

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 the entropy achieved by each $f' \in F'$, so we have that $|\Lambda| = |F'|$. Such calculation is performed as shown in Equation 4.

$$\Lambda = \{\lambda_1 = max(H(f'_1)), \lambda_2 = max(H(f'_2)), \dots, \lambda_M = max(H(f'_M))\} \tag{4}$$

In our proposed Entropy Difference Approach, such a metric is calculated twice, before and after we added to T'_+ a non evaluated instance $\hat{t}' \in \hat{T}'$.

3.3. Global Maximum Entropy Calculation

We denote as *global maximum entropy* γ the integral of the area under curve of the *local Entropy* Λ (previously defined in Section 3.2), as shown in Figure 2.

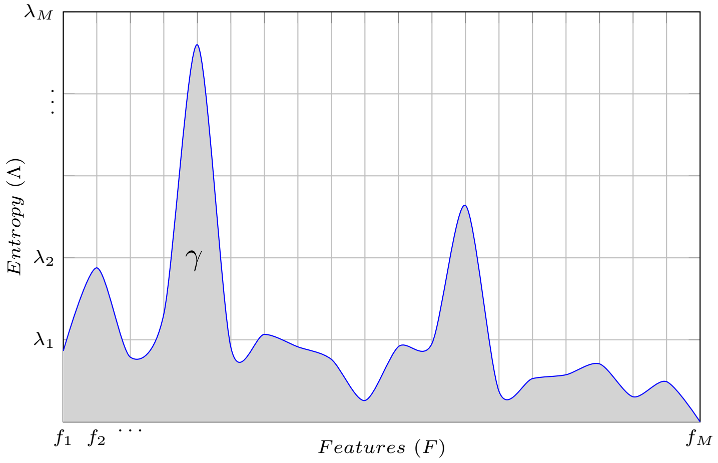


Figure 2: *Global Entropy* γ

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

$$\gamma = \int_{\lambda_1}^{\lambda_M} f(x)\,dx \approx \frac{\Delta x}{2} \sum_{n=1}^{|\Lambda|} \left(f(x_{n+1}) + f(x_n)\right)$$
$$\text{with} \tag{5}$$
$$\Delta x = \frac{(\lambda_M - \lambda_1)}{|\Lambda|}$$

The global entropy is a meta-feature that gives us information about the entropy achieved by all the features in T'_+, before and after we added to it a non evaluated instance. We use this information during the evaluation process, jointly with that given by Λ in Equation 4.

3.4. Entropy Difference Approach

Our proposed *Entropy Difference Approach* (*EDA*) is based on the Algorithm 3, which is able to evaluate and classify as reliable or unreliable a set of non evaluated (new) instances. It takes as input a set T'_+ of known non-default instances with features reduced, a set $\hat{T}'$ of non evaluated instances with the same features reduced and a previously trained threshold Θ. Then, it returns as output a set E, containing all the instances in $\hat{T}'$ classified as reliable or unreliable, depending on the Λ and γ information.

In step 3 of the algorithm, we calculate the Λ_a value, by using the reduced features from non-default instances only in T'_+, as described in Section 3.2, while in step 4 we obtain the global entropy γ (Section 3.3) in the same set. The steps from 5 to 23 process all the instances $\hat{t}' \in \hat{T}'$ from the instances to be classified with reduced features. After the calculation of Λ_b and γ_b values (steps 7 and 8) by adding the current instance $\hat{t}'$ to the set T'_+ of non-default instances with reduced features, the steps from 9 to 12 compare each $\lambda_a \in \Lambda_a$ with the corresponding feature $\lambda_b \in \Lambda_b$, counting how many times the value of λ_b is less or equal than λ_a. This is stored in a counter variable *count* (step 11). Steps 14-16 perform the same operation, but now it takes into account the global entropy γ comparisons. At the end of the previous sub-processes, in the steps from 17 to 21 we classify the current instance as reliable or unreliable, on the basis of the *count* value and the Θ threshold, then we set *count* to zero (step 22). The resulting set E is returned at the end of the entire process at step 24.

In this paper, we also include an evaluation of the computational complexity taken for the classification of a single instance $\hat{t}'$, because this information allows us determining the performance of our Algorithm 3 in a context of a real-time Credit Scoring system [61], a scenario where the response-time represents a primary aspect. We perform this operation by analyzing the theoretical complexity of the classification Algorithm 3, previously formalized. So, let N be the size of the set T'_+ (*i.e.*, $N = |T'_+|$) and M the size of the set F'_+ (*i.e.*, $M = |F'_+|$). The asymptotic time complexity of a single evaluation, in terms of *Big O notation*, can be determined on the basis of the following observations:

(i) as shown in Figure 3, the Algorithm 3 presents two nested loops given by the outer loop that starts at step 4 (L1 loop), which executes N times the inner loop L2 that starts at step 7 and other operations (*i.e.*, *getLo-*

Algorithm 3 *Entropy DifferenceApproach* (*EDA*)

Input: T'_{+}=Non-default instances with features reduced (see Section 3.1); $\hat{T}'$=Instances to evaluate with reduced features (see Section 3.1); Θ=Threshold

Output: E=Set of classified instances

```
procedure InstancesEvaluation($T'_{+}$,$\hat{T}'$, $\Theta$)
    $F'_{+} \leftarrow getAllFeatures(T'_{+})$
    $\Lambda_a \leftarrow getLocalMaxEntropy(F'_{+})$
    $\gamma_a \leftarrow getGlobalMaxEntropy(\Lambda_a)$
    for each $\hat{t}'$ in $\hat{T}'$ do
        $\hat{f}' \leftarrow getAllFeatures(\hat{t}')$
        $\Lambda_b \leftarrow getLocalMaxEntropy(F'_{+} + \hat{f}')$
        $\gamma_b \leftarrow getGlobalMaxEntropy(\Lambda_b)$
        for each $\lambda$ in $\Lambda$ do
            if $\lambda_b \leq \lambda_a$ then
                $count \leftarrow count + 1$
            end if
        end for
        if $\gamma_b \leq \gamma_a$ then
            $count \leftarrow count + 1$
        end if
        if $count > \Theta$ then
            $E \leftarrow (\hat{t}, reliable)$
        else
            $E \leftarrow (\hat{t}, unreliable)$
        end if
        $count \leftarrow 0;$
    end for
    return $E$
end procedure
```

calMaxEntropy, *getGlobalMaxEntropy*, plus comparisons and assignations operations, respectively with complexity $O(N)$, $O(M)$, $O(1)$, and $O(1)$);

(ii) the inner loop L2 executes M times operations of comparisons and assignations, respectively with complexity $O(1)$ and (1); and

(iii) the complexity related to the other operations executed by the algorithm (*i.e.*, *getLocalMaxEntropy*, *getGlobalMaxEntropy* in steps 2 and 3) is, respectively, $O(N)$ and $O(M)$.

The aforementioned considerations allow us determining that the asymptotic time complexity of the proposed algorithm is $O(N \times M)$, a complexity that can be effectively reduced by running in parallel the process over several machines, *e.g.*, by exploiting large scale distributed computing models such as *MapReduce* [62].

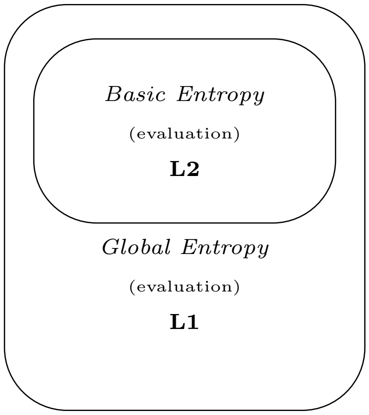


Figure 3: *Algorithm Nested Loops*

4. Experimental Setup

This section describes the datasets and metrics considered in the experiment, the adopted experiments methodology and implementation details of the state-of-the-art approach considered and the proposed approach.

4.1. Datasets

The datasets used during the experiments have been chosen for two reasons: first, they represent two benchmarks in this research field; second, they represent two different distributions of data (*i.e.*, unbalanced and slightly unbalanced). The first one is the *German Credit* (*GC*) dataset (unbalanced data distribution) and the second one is the *Australian Credit Approval* (*ACA*) dataset (slightly unbalanced data distribution). Both the datasets are freely available at the UCI Repository of Machine Learning Databases[1]. These datasets are released with all the attributes modified to protect the confidentiality of the data, and we used a version suitable for the algorithms that can not operate with categorical variables (*i.e.*, a version with all numeric attributes). It should be noted that, in case of other datasets that contain categorical variables, their conversion to numeric form is straightforward.

Table 1: Datasets Overview

Dataset	Total cases	Non-default	Default	Attributes	Classes
name	$\|T\|$	$\|T_+\|$	$\|T_-\|$	$\|F\|$	$\|C\|$
GC	$1,000$	700	300	21	2
ACA	690	307	383	15	2

The datasets' characteristics are summarized in Table 1 and detailed in the following:

German Credit (GC). It contains 1,000 instances: 700 of them are non-default instances (70.00%) and 300 are default instances (30.00%). Each instance is composed of 20 features, whose type is described in Table 2 and a binary class variable (reliable or unreliable).

[1] `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/`

Australian Credit Approval (ACA). It contains 690 instances, 307 of them are non-default instances (44.5%) and 383 are default instances (55.5%). Each instance is composed of 14 features and a binary class variable (reliable or unreliable). In order to protect the data confidentiality, all feature names and values of this dataset have been changed to meaningless symbols, as shown in Table 3, which reports the feature type instead of its description.

Table 2: Dataset GC Features

Feature	**Description**	**Feature**	**Description**
1	*Status of checking account*	*11*	*Present residence since*
2	*Duration*	*12*	*Property*
3	*Credit history*	*13*	*Age*
4	*Purpose*	*14*	*Other installment plans*
5	*Credit amount*	*15*	*Housing*
6	*Savings account/bonds*	*16*	*Existing credits*
7	*Present employment since*	*17*	*Job*
8	*Installment rate*	*18*	*Maintained people*
9	*Personal status and sex*	*19*	*Telephone*
10	*Other debtors/guarantors*	*20*	*Foreign worker*

Table 3: Dataset ACA Features

Feature	**Type**	**Feature**	**Type**
1	*Categorical field*	*8*	*Categorical field*
2	*Continuous field*	*9*	*Categorical field*
3	*Continuous field*	*10*	*Continuous field*
4	*Categorical field*	*11*	*Categorical field*
5	*Categorical field*	*12*	*Categorical field*
6	*Categorical field*	*13*	*Continuous field*
7	*Continuous field*	*14*	*Continuous field*

4.2. Metrics

This section introduces the metrics used to compare our proposed approach with the competitor in the experiments.

Accuracy. This metric reports the number of instances correctly classified and is calculated as:

$$Accuracy(\hat{T}) = \frac{|\hat{T}^{(+)}|}{|\hat{T}|}, \tag{6}$$

where $|\hat{T}|$ corresponds to the total number of instances, and $|\hat{T}^{(+)}|$ to the number of instances correctly classified.

Sensitivity. This metric measures the number of instances correctly classified as reliable, providing an important information since it allows evaluating the predictive power of our approach in terms of capability to identify the default cases. It is calculated as

$$Sensitivity(\hat{T}) = \frac{|\hat{T}^{(TP)}|}{|\hat{T}^{(TP)}| + |\hat{T}^{(FN)}|}, \tag{7}$$

where $|\hat{T}^{(TP)}|$ corresponds to the number of instances correctly classified as reliable and $|\hat{T}^{(FN)}|$ to the number of reliable instances erroneously classified as unreliable.

F-score. The *F-score* represents the weighted average of the *Precision* and *Recall* metrics and is considered an effective performance measure for unbalanced datasets [63]. Such a metric is calculated as

$$\begin{aligned} &F\text{-}score(T^{(P)}, T^{(R)}) = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recal} \\ &\text{with} \\ &Precision(T^{(P)}, T^{(R)}) = \frac{|T^{(R)} \cap T^{(P)}|}{|T^{(P)}|} \quad Recall(T^{(P)}, T^{(R)}) = \frac{|T^{(R)} \cap T^{(P)}|}{|T^{(R)}|}, \end{aligned} \tag{8}$$

where $T^{(P)}$ denotes the set of performed classifications of instances, and $T^{(R)}$ the set that contains the actual classifications of them.

Area Under the Receiver Operating Characteristic (AUC). This metric is a performance measure used to evaluate the effectiveness of a classification model [64, 65]. It is calculated as

$$\Theta(t_+, t_-) = \begin{cases} 1, & if\ t_+ > t_- \\ 0.5, & if\ t_+ = t_- \\ 0, & if\ t_+ < t_- \end{cases} \qquad AUC = \frac{1}{|T_+| \cdot |T_-|} \sum_{1}^{|T_+|} \sum_{1}^{|T_-|} \Theta(t_+, t_-), \tag{9}$$

where T_+ is the set of non-default instances, T_- is the subset default instances, and Θ indicates all possible comparisons between the instances of the two subsets T_+ and T_-. The final result is obtained by averaging all the comparisons.

4.3. Methodology, Competitors and Proposed Approach Implementation Details

The experiments have been performed using the *k-fold cross-validation*, with *k=10*. This approach allows us reducing the impact of data dependency, improving the reliability of the results. For this setup, we choose the Random Forest classifier [66] and three Naive Bayes improved classifiers [67, 68, 69] as competitors.

The *Random Forests* [66] approach represents one of the most common and powerful state-of-the-art techniques used for the Credit Scoring tasks, since in most of the cases it outperforms the other ones [10, 11, 12]. It consists of an ensemble learning approach for classification and regression based on the construction of a number of randomized decision trees during the training phase. The conclusion is inferred by averaging the obtained results and this technique can be used to solve a wide range of prediction problems. Naive Bayes classifiers use the Bayes Theorem by predicting probabilities that the input data belongs to a particular class. Thus, the class with the highest probability is considered the most likely class. We also included in the experiments this kind of classifier as competitor as it was also used for a similar problem before [38]. Therefore, we choose to also compare the proposed approach with some improved naive Bayes algorithms: Hidden Naive Bayes [67] (we will refer to this competitor as HNB), Deep Feature Weighted Naive Bayes [68] (we will refer to this competitor as $DFWNB$) and Correlation-based Feature Weighted Naive Bayes [69] (we will refer to this competitor as $CFWNB$). The implementation used to evaluate all the baselines performances in our experiments was the one made in the *Waikato Environment for Knowledge Analysis* (WEKA) machine learning package[2]. Parameters of these classifiers are shown in Table 4.

Table 4: Competitor Algorithms Parameters

Algorithm	Parameter	Values	Description
RF	*bagSizePercent*	100	Size of each bag as a percentage of the training set size
	batchSize	100	The preferred number of instances to process if batch prediction is being performed
	maxDepth	*Unlimited*	The maximum depth of the tree
	numIterations	100	The number of iterations to be performed
	numDecimalPlaces	2	The number of decimal places to be used for the output of numbers in the model
	seed	1	The random number seed to be used
HNB	*batchSize*	100	The preferred number of instances to process if batch prediction is being performed
	numDecimalPlaces	2	The number of decimal places to be used for the output of numbers in the model
$DFWNB$	*bagSizePercent*	50	Size of each bag as a percentage of the training set size
	batchSize	100	The preferred number of instances to process if batch prediction is being performed
	classifier	$DFWNB$	The base classifier to be used
	ignoreBelowDepth	0	Set to zero weight the attributes below this depth in the trees (0=disable)
	numBaggingIterations	10	Number of bagging iterations
	useCFSBasedWeighting	*True*	Use CFS-Based Feature Weighting
	useGainRatioBasedWeighting	*False*	Use Gain-Ratio-Based Weighting
	useInfoGainBasedWeighting	*False*	Use Info-Gain-Based Weighting
	useCFSBasedWeighting	*True*	Use CFS-Based Weighting
	useLogDepthWeighting	*False*	Use Log-Depth Weighting
	usePrunedTrees	*False*	Use Pruned Trees for bagging
	useReliefBasedWeighting	*False*	Use Relief-Based Weighting
	useZeroOneWeights	*False*	Use Zero-One Weights
	numDecimalPlaces	2	The number of decimal places to be used for the output of numbers in the model
	seed	1	The random number seed to be used
$CFWNB$	*batchSize*	100	The preferred number of instances to process if batch prediction is being performed
	numDecimalPlaces	2	The number of decimal places to be used for the output of numbers in the model

[2] `https://www.cs.waikato.ac.nz/ml/`

The proposed approach was developed in Java. The entropy measures needed for the approach presented in this paper have been developed by using *JavaMI*[3], a Java port of *MIToolbox*[4].

5. Experiments

In this section, we start the discussion about the experimental results. We divide this section into two subsections: in the first subsection, we present the experiments done to find the parameters of the proposed approach. Then, we discuss the final experiments results, comparing the proposed approach against its version without feature selection and also the competitors in real-world credit scoring datasets.

5.1. Parameter Tuning Experiments

In this Subsection, we discuss experiments results that helped us to find the best parameters of the proposed approach. In Section 5.1.1, we show how we found the features to be removed in our proposed approach using the feature selection step. Then, in Section 5.1.2, we report the experiments done that helped us to find the EDA threshold of our proposed approach.

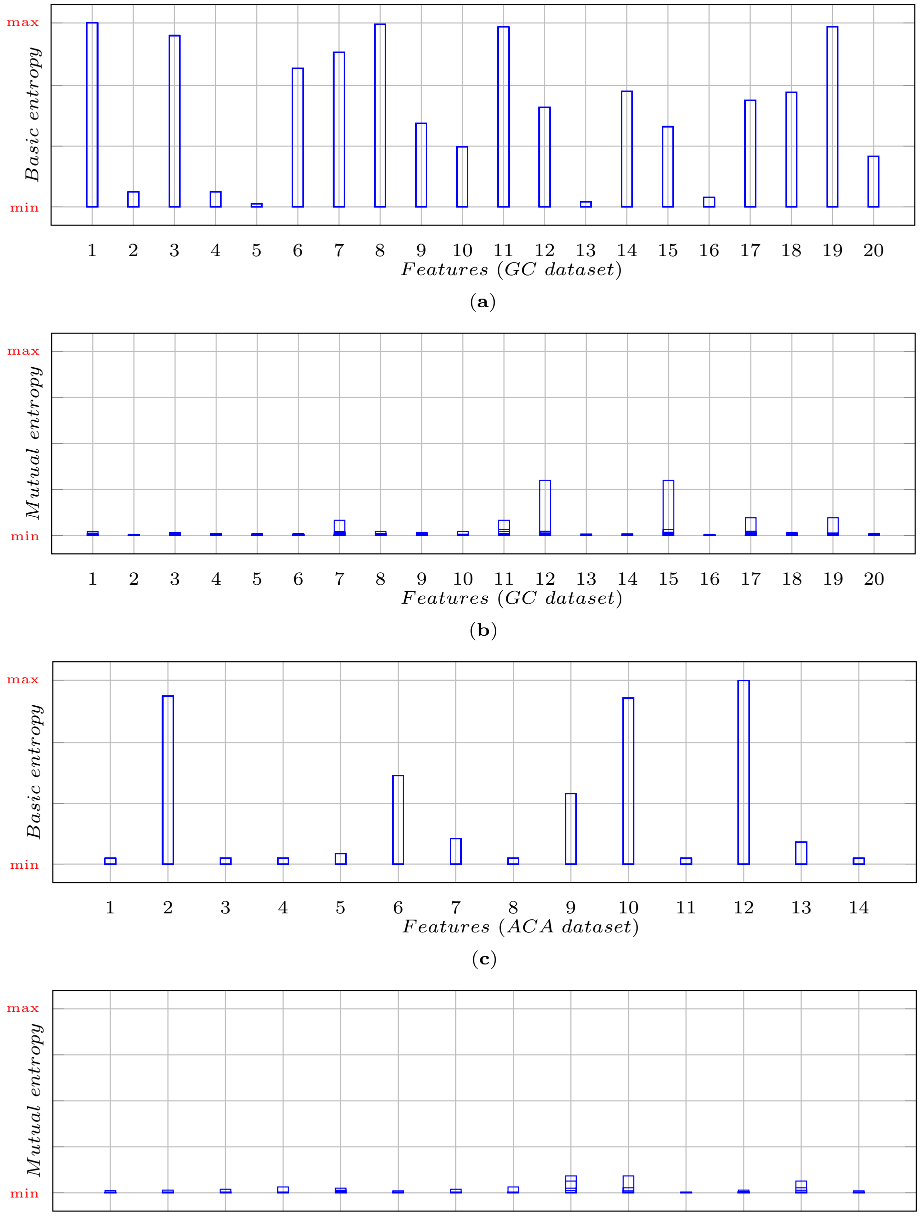
5.1.1. Feature Selection

In our first experiment to find parameters, we perform a study aimed at evaluating the contribution of each instance features in the proposed approach for the classification process. We do this by exploiting two different approaches of evaluation based on concepts of entropy previously discussed in Section 3.1. Results of each feature's basic and mutual entropies are shown in Figure 4.

The results shown in Figure 4 indicate that, although several features present a high level of entropy (*i.e.*, a low level of instance characterization, since the entropy increases as the data becomes equally probable), they have a positive

[3] `http://www.cs.man.ac.uk/~pococka4/JavaMI.html`

[4] `http://www.cs.man.ac.uk/~pococka4/MIToolbox.html`



(**a**)

(**b**)

(**c**)

(**d**)

Figure 4: *Basic and Mutual Entropy*

contribution in a mutual relation with other features (the number of mutual relations are represented through the horizontal lines in the feature bars). Considering that all values of entropy have been normalized in the $[0, 1]$ range (y-axis) and high values of Basic Entropy indicate high levels of uncertainty while high values of Mutual Entropy indicate large reductions of uncertainty, we can do the following considerations:

(i) The Basic Entropy results, reported in Figure 4.a and Figure 4.c, show that there are many features in the *GC* dataset that present a high level of Basic Entropy (*i.e.*, we considered as relevant value of Basic Entropy a value above the two thirds of the interval, *e.g.*, the features 1, 3, 6, 7, 8, 11, and 19 in *GC* dataset, as well as features 2, 10, and 12 in the *ACA* dataset).

(ii) The Mutual Entropy results, reported in Figure 4.b and Figure 4.d, show if there are features with a high level of Basic Entropy for which a Mutual Entropy with other features that reduces their uncertainty exists (we considered as relevant value of Mutual Entropy a value above the one quarter of the interval). In our case, there are not features that present such a status, since the features with a relevant value of Mutual Entropy are only the features 12 and 15 of the *GC* dataset, and there are no relevant features in the *ACA* dataset.

(iii) Such a scenario leads us towards the decision to exclude from the model definition process all the features with a high level of Basic Entropy, *i.e.*, the features 1, 3, 6, 7, 8, 11, and 19 of the *GC* dataset, and the feature 2, 10, and 12 of the *ACA* dataset. It should be noted that the high level of uncertainty reported by the Basic Entropy can be determined by two factors: either the information gathered by the system are inadequate or the nature of information has a low relevance for the classification task.

Furthermore, it should be observed how the aforementioned process reduces the computational complexity, since after the feature selection we excluded from the model definition process 7,000 elements (feature values involved in the evaluation process), *i.e.*, 35.00% of the total elements from the *GC* dataset and 2,070 elements (21,00% of the total elements) from the *ACA* dataset, as reported in Table 5.

Table 5: Feature Selection Process

Dataset name	Dataset total features	Removed total features	Processed total features	Reduction percentage
GC	20,000	7,000	13,000	35.00
ACA	9,660	2,070	7,590	21.00

5.1.2. Finding the Optimal EDA Threshold

According to the formalization of our approach made by the Algorithm 3, we need to define an optimal threshold Θ, that can be considered a function of the hyper-plane that will classify the samples $\hat{T}'$ into reliable or unreliable. Such an operation was performed by testing all the possible values, as shown in Figure 5. The tests were stopped as soon as the measured accuracy did not improve further and the obtained results showed that the optimal threshold Θ (*i.e.*, that related to the maximum value of Accuracy) was 3 for the *GC* dataset (Accuracy 70.30%) and 5 for the *ACA* dataset (Accuracy 67.20%).

5.2. Results

The experimental results are divided into three parts: (i) studying the effect of feature selection in the proposed approach; (ii) performance evaluation in public datasets; and (iii) performance under different levels of class unbalancing. We discuss these experiments in details in the following subsections.

5.2.1. The Effect of Feature Selection

We first report the experiment results of comparing the proposed approach with and without feature selection, a dataset preprocessing step of our approach discussed in Section 3.1. Figure 6 shows that removing features detected through

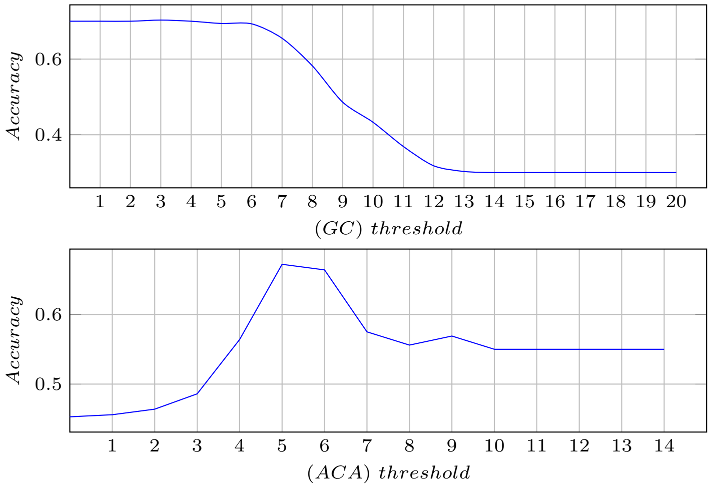


Figure 5: *Entropy Difference Approach Tuning*

the process performed in Section 5.1.1 presents a twofold advantage. First, it reduces the computational complexity, since fewer elements are involved in the evaluation process (as reported in Table 5). Second, it improves the performance in terms of all the metrics taken into account. From Figure 6, it may be highlighted the big jump in Sensitivity for both datasets (0.88 to 0.92 in GC dataset, and 0.70 to 0.86 in ACA dataset), showing that the proposed approach eliminates noise in the samples of the default class, increasing their classification.

5.2.2. Real-World Credit Scoring

We show the results considering different metrics that compare our proposed approach against the competitors in Figures 7 and 8. These figures show that our approach has promising results if compared with other baselines, even without any knowledge about default cases in its training step. The leftmost part of Figure 7 shows that our approach showed the best accuracy result for the most unbalanced dataset (GC), but not the best one for the slightly unbalanced dataset (ACA). However, in the rightmost part of Figure 7, it is shown that the proposed approach had the best default detection (sensitivity) for both datasets, with an almost perfect detection of GC default cases. The performance differ-

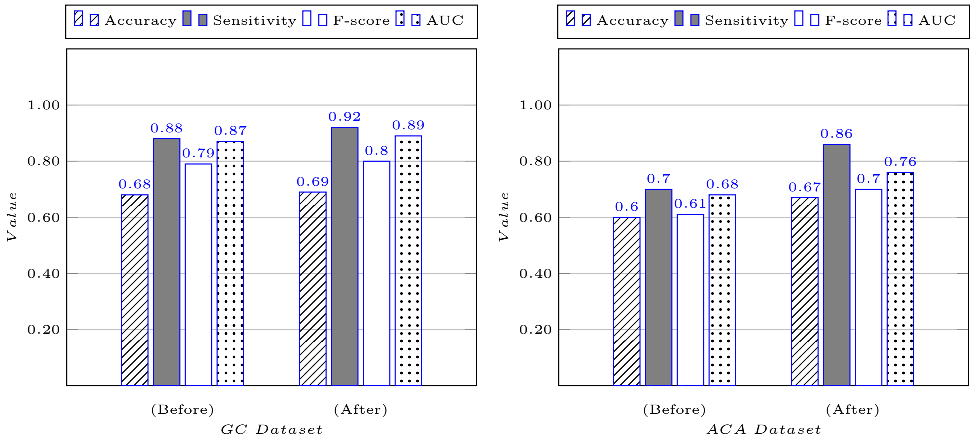


Figure 6: Proposed *EDA* approach metrics before and after the feature selection process.

ences in different datasets happens because of the complexities of samples in the ACA dataset, that are composed of different features from the ones in the GC dataset. The best baseline, namely a deep naive Bayes classifier [68] (DFWNB), succeeded only for the most balanced dataset (ACA), highlighting the fact that it performs an efficient credit scoring only when it has sufficient samples of both classes for training.

Figure 8 shows in its leftmost part that the f-score of our approach for the GC dataset is the best. The AUC of the proposed approach (rightmost part of Figure 8) is also the best for the GC dataset. All the other baselines (RF, HNB, DFWNB, CFWNB) had poor performances in this scenario, even considering the more balanced dataset (ACA). A special case that we would like to mention is about the low performance of the RF classifier, which is an ensemble of decision trees that is expected to work better in this real-world scenario. Such findings allow us to conclude that, at most, the baselines can have competitive performances against our approach only when balanced classes are available for training. We further show in the next subsection how our approach works better when less default cases in the training data are available.

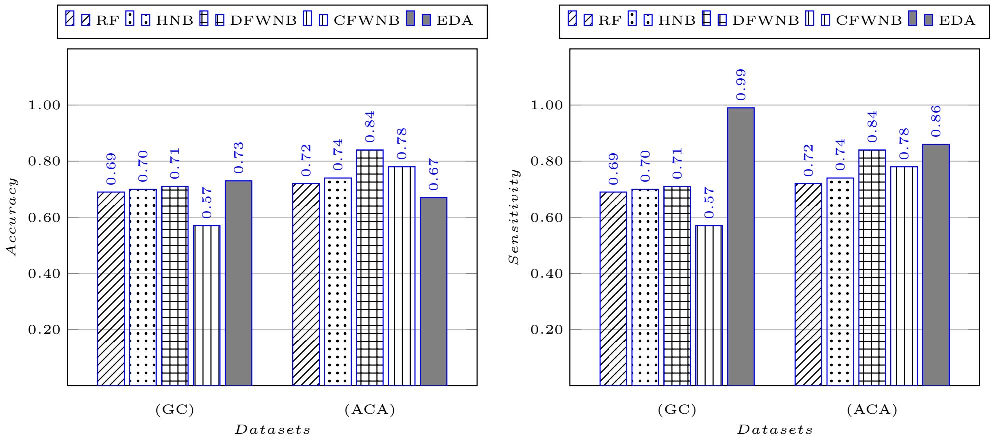


Figure 7: *Accuracy and Sensitivity Performance*

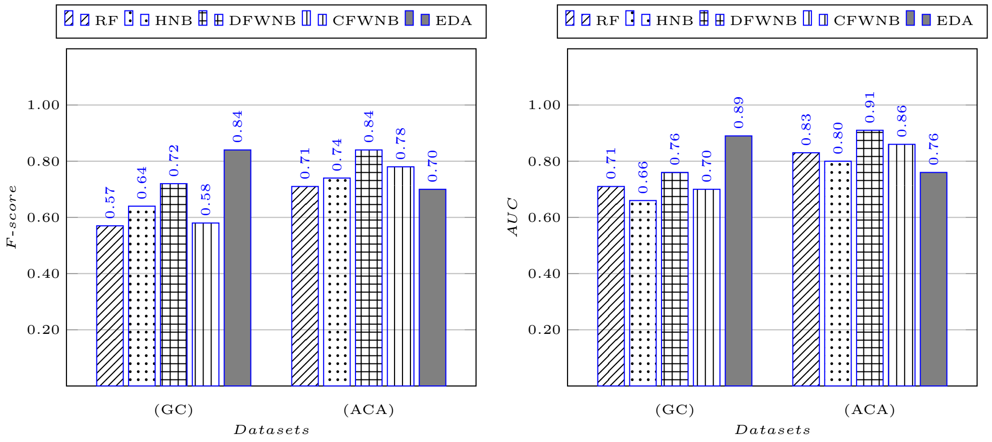


Figure 8: *F-score and AUC Performance*

5.2.3. Performance with Different Levels of Unbalance

In addition to the experiments discussed before, we tested our approach and competitors in the *GC* dataset (the most unbalanced one) with different unbalance levels. Therefore, we reduced the 300 original unreliable cases that compose the *GC* dataset according to five different levels of unbalance. In more detail, we used 50, 100, 150, 200, and 300 (original dataset) unreliable cases, joining them with the 700 reliable cases already present in the GC dataset. This creates new datasets with 750, 800, 850, 900 and 1000 (original dataset) samples, respectively. Therefore, the unreliable cases now correspond, respectively, to 6.66%, 12.50%, 17.64%, 22.22% and 30.00% of reliable cases in these datasets. For the experiments, we split the resulting datasets in training and test sets according to the 10-fold cross validation criterion, with our approach being the only one that does not consider default cases for training. Figure 9 shows the results of such experiments.

The results showed in Figure 9 highlight the proactive nature of our approach. By not considering the default cases in the training set, the imbalanced nature of such a problem does not influence our training. All the other approaches are influenced by the fact that less default training data is present, so they were able to reach accuracy comparable or better than ours only when more allowed default training data were available, as can be seen in the first row of Figure 9 (from 17.64% to 30%). However, the sensitivities of these approaches are still low as the training data is still unbalanced for the default cases, while our approach keep an almost perfect sensitivity in all unbalanced scenarios, as can be seen in the second row of Figure 9. The F-score metric of our approach, which is a recommendable measure for unbalanced environments, also highlights the proactive feature of our approach as it defeats all the baselines in all unbalanced scenarios, as can be seen in the third row of Figure 9. Finally, the fact that our approach defeats the baselines in 17 out of 20 experiments performed here further enriches the contributions of our approach to be applied in the unbalanced environment of credit scoring.

As found out in the previous experiments, we also realized that the DFWNB was the best competitor for this experiment. However, it is noticeable that it is biased when high levels of unbalance come into the game, a scenario that is more likely to happen in real world credit scoring datasets. Such an approach could defeat our approach in only two experiments in this subsection, but was the best one in just one experiment (AUC of 6.66% dataset, leftmost part of fourth row in Figure 9). Notwithstanding, with its good results, we believe that both approaches can be fused for a better credit scoring. This can be done, for example, by applying different weights for decisions of these different classifiers.

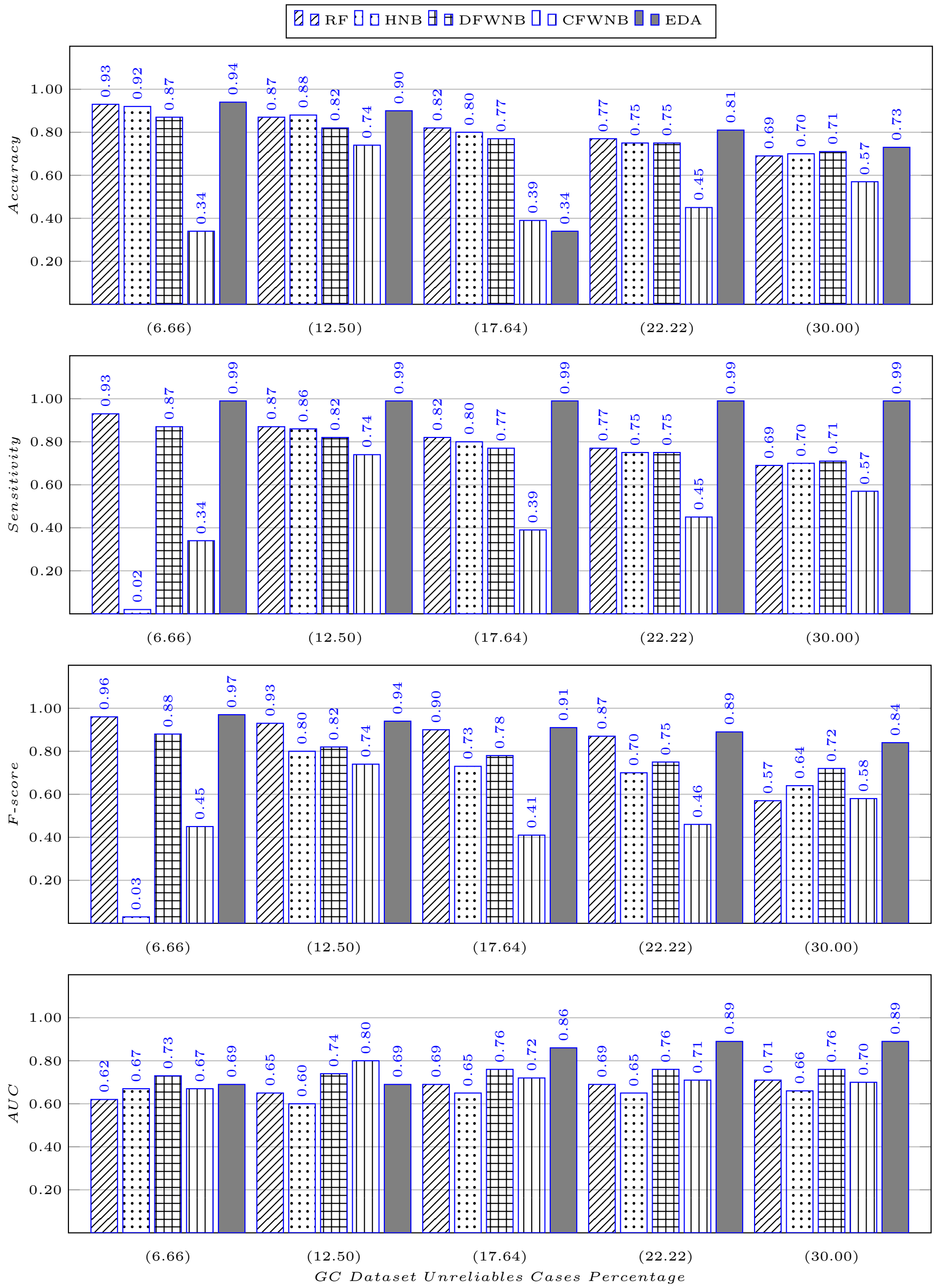


Figure 9: *Performance with Different Levels of Unbalance*

6. Conclusions and Future Work

The Credit Scoring machine learning 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, reducing therefore the losses due to unreliable users. However, several issues are found in such an application, such as the data imbalance problem in datasets, where the number of unreliable cases are quite smaller than the number of reliable cases, and also the cold-start problem, where there is scarcity or absence of non-reliable previous cases. These issues can seriously affect machine learning approaches aimed at classification of new instances in the Credit Score environment.

This paper proposes a novel approach of Credit Scoring that exploits entropy-based criteria in order to build a model able to classify a new instance without the knowledge of past non-reliable instances. Our approach works by comparing the entropy behavior of existing reliable samples before and after adding an instance under investigation. This way, our approach can operate in a proactive manner, facing the cold-start and the data imbalance problems that reduce the effectiveness of the canonical approaches of Credit Scoring. The experimental results underline two main aspects related to our approach: one the one hand, it has competitive performances if compared to existing classifiers when the training set is composed of slightly unbalanced (or almost balanced) classes; on the other hand it is able to outperform its competitors specially when the training process is characterized by an unbalanced distribution of training data. This last aspect represents an important result, since it shows the capability of the proposed approach to operate in scenarios where the canonical approaches of machine learning are not able to achieve optimal performance. This is especially true in the typical contexts of Credit Scoring, where an unbalanced distribution of data is usually present. Even without totally replacing the canonical approaches of Credit Scoring, our approach offers the possibility to overcome the cold-start issue, together with the capability to manage the unbalanced distribution of data, giving the opportunity to be jointly used with existing approaches and thus resulting in an effective hybrid model.

According to the previous considerations, a direction of future work where we are headed to is to evaluate the advantages and disadvantages related to the inclusion of the default cases in the model definition process, as well as the evaluation of our approach in heterogeneous scenarios that involve different types of financial data, such as those generated by an electronic commerce environment. A final goal is then to define a novel approach (hybrid or only based on the proposed approach) able to operate in all possible scenarios, effectively.

Acknowledgments

The research performed in this paper has been supported by the "Bando Aiuti per progetti di Ricerca e Sviluppo - POR FESR 2014-2020 - Asse 1, Azione 1.1.3. Project IntelliCredit: AI-powered digital lending platform". We also would like to thank the authors of the Deep Feature Weighted Naive Bayes approach, who kindly shared with us their source code.

References

[1] J. Morrison, Introduction to survival analysis in business, The Journal of Business Forecasting 23 (1) (2004) 18.

[2] L. J. Mester, et al., Whats the point of credit scoring?, Business review 3 (1997) 3–16.

[3] W. E. Henley, Statistical aspects of credit scoring., Ph.D. thesis, Open University (1994).

[4] W. Henley, et al., Construction of a k-nearest-neighbour credit-scoring system, IMA Journal of Management Mathematics 8 (4) (1997) 305–321.

[5] A. Fensterstock, Credit scoring and the next step, Business Credit 107 (3) (2005) 46–49.

[6] J. Brill, The importance of credit scoring models in improving cash flow and collections, Business Credit 100 (1) (1998) 16–17.

[7] A. D. Pozzolo, O. Caelen, Y. L. Borgne, S. Waterschoot, G. Bontempi, Learned lessons in credit card fraud detection from a practitioner perspective, Expert Syst. Appl. 41 (10) (2014) 4915–4928. doi:10.1016/j.eswa.2014.02.026.

[8] G. E. Batista, R. C. Prati, M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 20–29.

[9] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intelligent Data Analysis 6 (5) (2002) 429–449.

[10] S. Lessmann, B. Baesens, H. Seow, L. C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, European Journal of Operational Research 247 (1) (2015) 124–136.

[11] I. Brown, C. Mues, An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl. 39 (3) (2012) 3446–3453. doi:10.1016/j.eswa.2011.09.033.

[12] S. Bhattacharyya, S. Jha, K. K. Tharakunnel, J. C. Westland, Data mining for credit card fraud: A comparative study, Decision Support Systems 50 (3) (2011) 602–613. doi:10.1016/j.dss.2010.08.008.
URL http://dx.doi.org/10.1016/j.dss.2010.08.008

[13] R. Saia, S. Carta, An entropy based algorithm for credit scoring, in: A. M. Tjoa, L. D. Xu, M. Raffai, N. M. Novak (Eds.), Research and Practical Issues of Enterprise Information Systems - 10th IFIP WG 8.9 Working Conference, CONFENIS 2016, Vienna, Austria, December 13-14, 2016, Proceedings, Vol. 268 of Lecture Notes in Business Information Processing, 2016, pp. 263–276. doi:10.1007/978-3-319-49944-4_20.
URL http://dx.doi.org/10.1007/978-3-319-49944-4_20

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

[15] M. Doumpos, C. Zopounidis, Credit scoring, in: Multicriteria Analysis in Finance, Springer, 2014, pp. 43–59.

[16] R. Saia, S. Carta, Introducing a vector space model to perform a proactive credit scoring, in: International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, Springer, 2016, pp. 125–148.

[17] R. Saia, S. Carta, D. R. Recupero, G. Fenu, M. Saia, A discretized enriched technique to enhance machine learning performance in credit scoring, in: KDIR, 2019.

[18] R. Saia, S. Carta, G. Fenu, A wavelet-based data analysis to credit scoring, in: Proceedings of the 2nd International Conference on Digital Signal Processing, ACM, 2018, pp. 176–180.

[19] R. Saia, S. Carta, A fourier spectral pattern analysis to design credit scoring models, in: Proceedings of the 1st International Conference on Internet of Things and Machine Learning, ACM, 2017, p. 18.

[20] R. Saia, S. Carta, A linear-dependence-based approach to design proactive credit scoring models., in: KDIR, 2016, pp. 111–120.

[21] V. Ceronmani Sharmila, K. Kumar R, S. R, S. D, H. R, Credit card fraud detection using anomaly techniques, in: International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1–6.

[22] F. Fang, Y. Chen, A new approach for credit scoring by directly maximizing the kolmogorovsmirnov statistic, Computational Statistics & Data Analysis 133 (2019) 180 – 194.

[23] X. Zhang, Y. Yang, Z. Zhou, A novel credit scoring model based on optimized random forest, in: IEEE Annual Computing and Communication Workshop and Conference (CCWC), 2018, pp. 60–65.

[24] S. Maldonado, G. Peters, R. Weber, Credit scoring using three-way decisions with probabilistic rough sets, Information Sciencesdoi:https://doi.org/10.1016/j.ins.2018.08.001.
URL http://www.sciencedirect.com/science/article/pii/S0020025518306078

[25] B. Zhu, W. Yang, H. Wang, Y. Yuan, A hybrid deep learning model for consumer credit scoring, in: International Conference on Artificial Intelligence and Big Data (ICAIBD), 2018, pp. 205–208. doi:10.1109/ICAIBD.2018.8396195.

[26] Y. Tian, Z. Yong, J. Luo, A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines, Applied Soft Computing 73 (2018) 96 – 105.

[27] V. Neagoe, A. Ciotec, G. Cucu, Deep convolutional neural networks versus multilayer perceptron for financial prediction, in: International Conference on Communications (COMM), 2018, pp. 201–206.

[28] S. Ali, K. A. Smith, On learning algorithm selection for classification, Appl. Soft Comput. 6 (2) (2006) 119–138. doi:10.1016/j.asoc.2004.12.002.

[29] D. J. Hand, Measuring classifier performance: a coherent alternative to the area under the ROC curve, Machine Learning 77 (1) (2009) 103–123. doi:10.1007/s10994-009-5119-5.

[30] T.-S. Lee, I.-F. Chen, 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.

[31] G. Wang, J. Hao, J. Ma, H. Jiang, A comparative assessment of ensemble learning for credit scoring, Expert Syst. Appl. 38 (1) (2011) 223–230. doi:10.1016/j.eswa.2010.06.048.

[32] N.-C. Hsieh, Hybrid mining approach in the design of credit scoring models, Expert Systems with Applications 28 (4) (2005) 655–665.

[33] J. Lpez, S. Maldonado, Profit-based credit scoring based on robust optimization and feature selection, Information Sciences 500 (2019) 190 – 202.

[34] S. Guo, H. He, X. Huang, A multi-stage self-adaptive classifier ensemble model with application in credit scoring, IEEE Access 7 (2019) 78549–78559.

[35] H. Zhang, H. He, W. Zhang, Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring, Neurocomputing 316 (2018) 210 – 221.

[36] X. Feng, Z. Xiao, B. Zhong, J. Qiu, Y. Dong, Dynamic ensemble classification for credit scoring using soft probability, Applied Soft Computing 65 (2018) 139 – 151. doi:https://doi.org/10.1016/j.asoc.2018.01.021. URL http://www.sciencedirect.com/science/article/pii/S1568494618300279

[37] D. Tripathi, D. R. Edla, V. Kuppili, A. Bablani, R. Dharavath, Credit scoring model based on weighted voting and cluster based feature selection, Procedia Computer Science 132 (2018) 22 – 31, international Conference on Computational Intelligence and Data Science.

[38] R. Vedala, B. R. Kumar, An application of naive bayes classification for credit scoring in e-lending platform, in: International Conference on Data Science Engineering (ICDSE), 2012, pp. 81–84. doi:10.1109/ICDSE.2012.6282321.

[39] D. Sewwandi, K. Perera, S. Sandaruwan, O. Lakchani, A. Nugaliyadde, S. Thelijjagoda, Linguistic features based personality recognition using so-

cial media data, in: 2017 6th National Conference on Technology and Management (NCTM), 2017, pp. 63–68. doi:10.1109/NCTM.2017.7872829.

[40] X. Sun, B. Liu, J. Cao, J. Luo, X. Shen, Who am i? personality detection based on deep learning for texts, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6.

[41] R. F. López, J. M. Ramon-Jeronimo, Modelling credit risk with scarce default data: on the suitability of cooperative bootstrapped strategies for small low-default portfolios, JORS 65 (3) (2014) 416–434. doi:10.1057/jors.2013.119.
URL http://dx.doi.org/10.1057/jors.2013.119

[42] G. Garibotto, P. Murrieri, A. Capra, S. D. Muro, U. Petillo, F. Flammini, M. Esposito, C. Pragliola, G. D. Leo, R. Lengu, N. Mazzino, A. Paolillo, M. D'Urso, R. Vertucci, F. Narducci, S. Ricciardi, A. Casanova, G. Fenu, M. D. Mizio, M. Savastano, M. D. Capua, A. Ferone, White paper on industrial applications of computer vision and pattern recognition, in: ICIAP (2), Vol. 8157 of Lecture Notes in Computer Science, Springer, 2013, pp. 721–730.

[43] A. Chatterjee, A. Segev, Data manipulation in heterogeneous databases, ACM SIGMOD Record 20 (4) (1991) 64–68.

[44] B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the cold start problem in recommender systems, Expert Syst. Appl. 41 (4) (2014) 2065–2073. doi:10.1016/j.eswa.2013.09.005.
URL http://dx.doi.org/10.1016/j.eswa.2013.09.005

[45] L. H. Son, Dealing with the new user cold-start problem in recommender systems: A comparative review, Inf. Syst. 58 (2016) 87–104. doi:10.1016/j.is.2014.10.001.
URL http://dx.doi.org/10.1016/j.is.2014.10.001

[46] I. Fernández-Tobías, P. Tomeo, I. Cantador, T. D. Noia, E. D. Sciascio, Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback, in: S. Sen, W. Geyer, J. Freyne, P. Castells (Eds.), Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, ACM, 2016, pp. 119–122. doi:10.1145/2959100.2959175.
URL http://doi.acm.org/10.1145/2959100.2959175

[47] J. Attenberg, F. J. Provost, Inactive learning?: difficulties employing active learning in practice, SIGKDD Explorations 12 (2) (2010) 36–41. doi:10.1145/1964897.1964906.
URL http://doi.acm.org/10.1145/1964897.1964906

[48] V. Thanuja, B. Venkateswarlu, G. Anjaneyulu, Applications of data mining in customer relationship management, Journal of Computer and Mathematical Sciences Vol 2 (3) (2011) 399–580.

[49] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284. doi:10.1109/TKDE.2008.239.

[50] V. Vinciotti, D. J. Hand, Scorecard construction with unbalanced class sizes, Journal of Iranian Statistical Society 2 (2) (2003) 189–205.

[51] A. I. Marqués, V. García, J. S. Sánchez, On the suitability of resampling techniques for the class imbalance problem in credit scoring, JORS 64 (7) (2013) 1060–1070. doi:10.1057/jors.2012.120.
URL http://dx.doi.org/10.1057/jors.2012.120

[52] S. F. Crone, S. Finlay, Instance sampling in credit scoring: An empirical study of sample size and balancing, International Journal of Forecasting 28 (1) (2012) 224–238.

[53] L. Jiang, C. Qiu, C. Li, A novel minority cloning technique for cost-sensitive learning, International Journal of Pattern Recognition and Artificial Intelligence 29 (04) (2015) 1551004.

[54] L. Jiang, C. Li, S. Wang, Cost-sensitive bayesian network classifiers, Pattern Recognition Letters 45 (2014) 211 – 216.

[55] B. Tang, H. He, Gir-based ensemble sampling approaches for imbalanced learning, Pattern Recognition 71 (2017) 306 – 319.

[56] X. Yang, Q. Kuang, W. Zhang, G. Zhang, Amdo: An over-sampling technique for multi-class imbalanced problems, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1672–1685.

[57] J. Zhang, V. S. Sheng, Q. Li, J. Wu, X. Wu, Consensus algorithms for biased labeling in crowdsourcing, Information Sciences 382-383 (2017) 254 – 273.

[58] S. Vluymans, A. Fernández, Y. Saeys, C. Cornelis, F. Herrera, Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: a fuzzy rough set approach, Knowledge and Information Systems 56 (1) (2018) 55–84.

[59] Z. Zhang, B. Krawczyk, S. Garcia, A. Rosales-Perez, F. Herrera, Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data, Knowledge-Based Systems 106 (2016) 251 – 263.

[60] Y. Liu, M. Schumann, Data mining feature selection for credit scoring models, Journal of the Operational Research Society 56 (9) (2005) 1099–1108.

[61] J. R. Lent, M. Lent, E. R. Meeks, Y. Cai, T. J. Coltrell, D. W. Dowhan, Method and apparatus for real time on line credit approval, uS Patent 6,405,181 (Jun. 11 2002).

[62] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113. doi:10.1145/1327452.1327492.
URL http://doi.acm.org/10.1145/1327452.1327492

[63] A. D. Pozzolo, O. Caelen, R. A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification, in: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015, IEEE, 2015, pp. 159–166. doi:10.1109/SSCI.2015.33.
URL http://dx.doi.org/10.1109/SSCI.2015.33

[64] D. M. W. Powers, Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation, Journal of Machine Learning Technologies 2 (1) (2011) 37–63.

[65] D. Faraggi, B. Reiser, Estimation of the area under the roc curve, Statistics in medicine 21 (20) (2002) 3093–3106.

[66] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[67] L. Jiang, H. Zhang, Z. Cai, A novel bayes model: Hidden naive bayes, IEEE Transactions on Knowledge and Data Engineering 21 (10) (2009) 1361–1371. doi:10.1109/TKDE.2008.234.

[68] L. Jiang, C. Li, S. Wang, L. Zhang, Deep feature weighting for naive bayes and its application to text classification, Engineering Applications of Artificial Intelligence 52 (2016) 26–39.

[69] L. Jiang, L. Zhang, C. Li, J. Wu, A correlation-based feature weighting filter for naive bayes, IEEE Transactions on Knowledge and Data Engineering 31 (2) (2019) 201–213.