An Experimental Comparison of Credit Risk Classification

Yosimar O. Serrano-Silva[1], Yenny Villuendas-Rey[2], Cornelio Yáñez-Márquez[1]

[1]Centro de Investigación en Computación del Instituto Politécnico Nacional, Avenida Juan de Dios Bátiz esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, Gustavo A. Madero, CP 07738, CDMX, México.
[2]Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, Nueva Industrial Vallejo, Gustavo A. Madero, 07700 CDMX, México.

oswaldo17@live.com.mx, yenny.villuendas@gmail.com, coryanez@gmail.com.

Abstract. Credit is a fundamental aspect for finances, and there is the necessity of developing automated decision-making systems, which to some extent, can reduce the risk involved for the institutions granting credit. In this paper, we tested several supervised classification algorithms, and compare their performance over some well-known credit datasets, according to the Area under the ROC Curve.

Keywords: credit risk classification, supervised classification, imbalanced data.

1 Introduction

In commercial banking (business, personnel, etc.), Institutions assume some credit risk in every single asset operations that they perform (loans, lines of credit, guarantees, etc.) because on the one hand, this Institutions can never know everything about the customers and on the other, compliance with the payment obligations depends on events that nobody knows if could happened or not, that is, there is uncertainty about whether or not the customer will pay its debt. For those reasons, a lot of models have been proposed to evaluate credit risk [1].

In the literature, we can find a lot of different techniques that have been proposed to solve the problem of the credit risk and credit approval. Some approaches are based on the credit area, but address the issue of how credit managed badly can produce personal bankruptcy [2]. However other approaches try to solve some issues from credit rating systems using hidden Markov models [3] and others, based on the increasingly important role that social media has been playing sharing individual's opinions on many financial issues, analyzing whether these opinions can accurately predict credit risk[4].

Nevertheless uncertainty still exists in the financial area and there is the necessity of developing automated decision-making systems, which to some extent, can reduce the risk involved for the institutions granting credit or any of the operations previously mentioned.

Therefore, risk analysis, with all its factors and types, has some difficulties that nowadays we are still facing: How to integrate all these variables within an automated decision-making system? And after that, How to measure the significance of each variable and its contribution to the adoption of one decision or another?

In order to make a decision automatically, one solution is to use a model of pattern recognition to solve the classification task [5], i.e., from a certain dataset containing relevant information about customers that request a type of credit to a financial Institution, the classifier can help these Institutions to decide whether it is appropriate or not to grant the request.

However, one of the drawbacks when working with datasets that contain financial information is that these, in most cases, have missing values, unbalanced classes and have mixed attributes types[6], for which the classification model should be able to address this situation.

An unbalanced class is present in a dataset when one of the classes has more elements than the others. The problem working with this kind of datasets is that, in general, this situation creates biased learning. At the moment of testing this phenomenon used to give us inaccurate results about the performance of the classifiers due to the biased learning causes that the classifiers appropriately recognize only the elements of the ruling class.

In this article an experimental studio about the performance of different supervised classifiers with credit datasets is presented, which in most of the cases has missing values, mixed attribute types and unbalanced classes.

The rest of the paper is organized as follows. Section 2 describes in detail some aspects of the classifiers used in this comparison and section 3 offers a discussion about the results obtained. Finally the paper ends with some conclusions and future research suggestions.

2 Sampling and error measurement

For the purpose of validating the performance of a classifier the most common method applied in the literature is stratified cross-validation (SCV). This technique involves partitioning the dataset into two complementary subsets. The first subset is used for training the classifier and the second one is used for testing. It places an equal number of patterns from every class into each partition to keep the same class distributions.

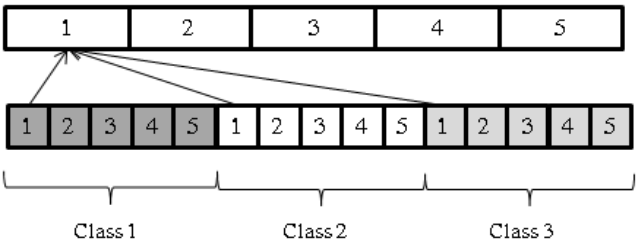

Fig. 1. Process to divide the dataset into k=5 subsets following the SCV technique.

To achieve that, as it shown in the Fig.1, firstly it is necessary to divide the complete data set into k different partitions, which in turn, are formed by one of the k partitions from the different classes.

After that, one of these subsets is taken as testing data and the remaining k-1 subsets are used as training data. This process is repeated k times and every subset is use as testing data exactly once.

As a result of the characteristics of the datasets from the financial and credit environment, it is necessary to choose a correct error measurement that can handle the problem of unbalanced classes and avoid inaccurate results. A metric that meet this requirement is the Area under the ROC curve (AUC) that is a popular classification metric which exhibits the benefit of being independent of the class distribution. The results of this measurement can be interpreted as follows: ideal classification model if the value of AUC is 1.0 and random classifier if the value obtained is 0.5. This measurement has been demonstrated that it can be calculated as the average of the True Positive Rate (TPR) and True negative Rate (TNR) for discrete classifiers by Sokolava et al.[7].

Table 1. Confusion matrix.

	Predicted as Positive	*Predicted as negative*
Positive instances	TP	FP
Negative instances	FN	TN

$$AUC = (TPR + TNR) / 2 \quad (1)$$

$$TPR = TP / (TP + TN) \quad (2)$$

$$TNR = TN / (TN + TP) \quad (3)$$

3 Results and discussion

3.1 Datasets

To carry out the different experiments, three datasets that belong to credit environments were used, which meet with the characteristics that are often present in this environment , it means that in the three datasets we can find missing values, data with hybrid types and unbalanced classes. These datasets were taken from the Machine Learning repository of the University of California [8].

Table 2. Caracteristics of the datasets used in this work.

Data set	*Instances*	*Attributes*	*Classes*	*Missing values*	*Unbalance Ratio*
Credit-australian	690	16	2	Yes	1.247
Credit-german	1000	21	2	No	2.333
Credit-approval	690	16	2	Yes	1.247

The German credit data corresponds to credit approvals. It has 1000 records with 20 attributes (7 numerical, 13 categorical) and do not have missing values. This dataset includes a cost matrix, due to the fact that it is considered worse to classify a client as good when is bad, than define a customer as bad when in fact is good.

The Credit Approval dataset contains 690 instances with 15 attributes (continuous and nominal) and presents some missing values. The purpose of this dataset is to predict whether an instance had a credit approved or not. The Australian credit dataset is a variation of the first one, used by the Statlog project [9].

3.2 Algorithms to compare.

Nearest Neighbor (1-NN).

Nearest Neighbor model [10] is part of the family of learning techniques called instance-based learning. The learning of this kind of algorithms is limited to stock in memory the patterns from the training set. This classifier is based on the idea that individuals from a population often share some similar properties and certain characteristics with the individuals around them. Thus, the classification of a pattern is carried out using the closest instances of the training set based on a dissimilarity measure. Due to the use of a distance measure, algorithms like this are called minimum distance classifiers.

One of the most popular similarity measures to numerical attributes is the Euclidean distance.

$$d(y,x) = \sqrt{\sum_{j=1}^{n}(y_j - x_j)^2} \quad (4)$$

C4.5

C4.5 algorithm builds decision trees from a dataset using the information entropy concept [11]. C4.5 chooses, at each node, the attribute of the pattern that splits effectively its set of samples into subsets improved in one class or the other. The criterion of splitting is the difference in entropy (normalized information gain). The attribute with the highest normalize information gain value is chosen to make the decision. Finally, this algorithm has three base cases:

- Instance of previously-unseen class encountered. The algorithm makes a decision node higher up the tree using the expected value.
- None of the features provide any information gain. Once more the algorithm makes a decision node higher up the tree using the expected value.
- The algorithm makes a decision node higher up the tree using the expected value. C4.5 creates a leaf node for the decision tree saying to choose the class.

Repeated Incremental Pruning to produce Error Reduction (RIPPER).

RIPPER is a classification algorithm that was proposed by William W Cohen [12]. It is based on association rules with reduced error pruning (REP), which is a common and effective technique of decision tree algorithms. In REP for rules algorithms, the training data is split into a growing set and a pruning set. First using some heuristic method, an initial rule set is formed (growing set). This overlarge rule set is then repeatedly simplified by applying one of a set of pruning operators typical pruning operators would be to delete any single condition or any single rule. At each stage of simplification, the pruning operator chosen is the one that yields the greatest reduction of error on the pruning set. Simplification ends when applying any pruning operator would increase error on the pruning set [12].

Multilayer Perceptron (MLP).

The Artificial Neural Network is a learning paradigm based on biological neural networks, in particular the human brain. Anatomically this system is composed for networks of biological neurons interconnected, which are able to process and conduct electrical impulses to produce an output. In 1943 it was proposed an abstract and simple model of an artificial neuron as a binary device [13]. This model has an operating threshold below which this neuron is inactive. Also, it has excitatory and inhibitory inputs, and depending on if there is any of these inputs the neuron is active.

This model is very simple, if there is not an inhibitory input, the resultant of the excitatory inputs is determined and if this is greater than the threshold, the output is 1 otherwise is 0 (Fig.2).

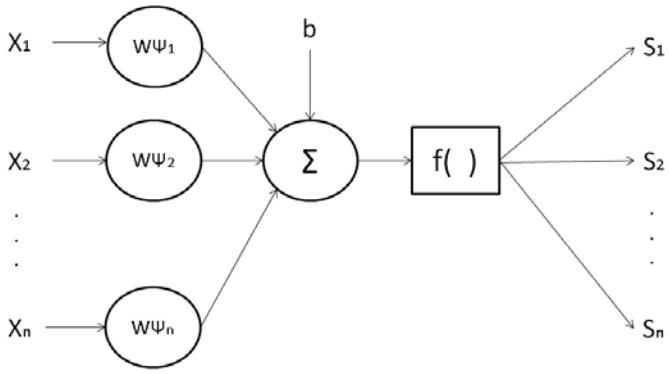

Fig. 2. Artificial neuron scheme.

Based on the work of McCulloch and Pitts, in 1953 it was proposed the perceptron [14]. One of the most interesting characteristics of this model was its ability of learning to recognize and classify objects. The perceptron was constituted by a set of input sensors which receives the patterns to recognize or classify and am output neuron to do the classification task. Nevertheless, this model was not capable to converge on good solutions in problems with classes linearly non-separable.[15].

Finally in 1986 the Multilayer Perceptron (MLP) [16] was proposed to solve the limitations of the perceptron. This network consists of multiple layers of artificial neurons; the most common architecture of a simple MLP network has 3 layers: an input and an output layer with one hidden layer however, the general model allows use an unlimited number of hidden layers.

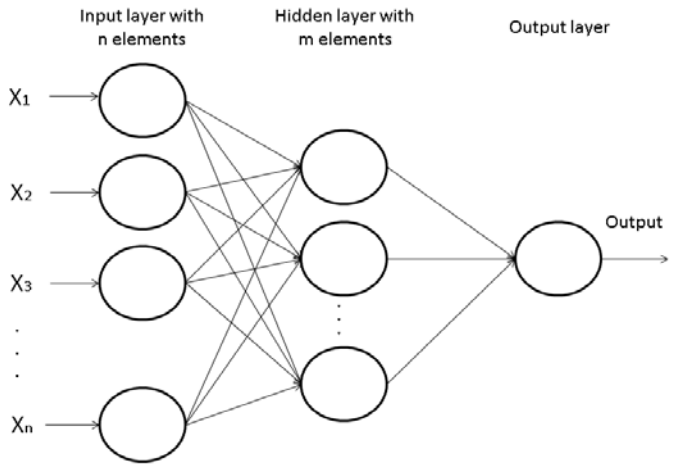

Fig. 3. General model of a MLP network with one hidden layer

Finally the supervised training stage is one of the most popular algorithms called back-propagation. The bases of this algorithm are in the error-correction learning rule [16].

Sequential Minimal Optimization Algorithm for training a Support Vector classifier (SMO).

Sequential Minimal Optimization (SMO) [17] is an algorithm for training Support Vector Machines [18] and was proposed to solve the problem of the very large quadratic programming optimization problem that implies this kind of training.

Considering a classification problem with a dataset (x_1, y_1),…, (x_n, y_n) where x_i is an input vector and y_i is a binary label corresponding to it. A soft-margin support vector machine is trained by solving a quadratic programming problem, which is expressed in the dual form as follows:

$$\underset{\infty}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j K(x_i, x_j) \alpha_i \alpha_j, \tag{5}$$

Subject to:

$$0 \leq \alpha_i \leq C, \quad for\ i = 1, 2, \ldots, n, \tag{6}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{7}$$

Where C is an SVM hyperparameter and $K(x_i, x_j)$ is the kernel function, both supplied by the user; and the variables α_i are Lagrange multipliers.

This is an iterative algorithm to solve the optimization problem. SMO converts this problem into a set of smallest possible sub-problems, which are then solved analytically. Due to the fact of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2 the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C \tag{8}$$

$$y_1 \alpha_1 + y_2 \alpha_2 = k \tag{9}$$

And this reduced problem can be solved analytically. The algorithm proceeds as follows [17]:

- Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
- Pick a second multiplier \alpha_2 and optimize the pair (α_1, α_2).
- Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved.

Naive Bayes (NB).

Naïve Bayes algorithm [19] assumes, for an instance x that its attributes x1, x2, …, xn have a conditional independence due to its class. For this reason the conditional likelihood of every attribute can be expressed as follows.

$$p(x|\omega_i) = \prod_{j=1}^{n} p(x_j|\omega_i) \quad (10)$$

Using the Bayes theorem, the posteriori likelihood is:

$$p(\omega_i|x) = \mathrm{p}(\omega_i) \prod_{j=1}^{n} p(x_j|\omega_i) \quad (11)$$

Finally, for every pattern of the testing set is given a class as is describe in the following equation

$$\omega^* = argmax_{\omega j} \mathrm{p}(\omega_i) \prod_{j=1}^{n} p(x_i|\omega_j) \quad (12)$$

Each was tested with the different datasets in Waikato Environment for Knowledge Analysis (WEKA) software [20] in its version number 3.6.13 using the default parameters offered.

3.3 Error Measurement.

The results obtained with the different models to every dataset, using the Stratified Cross Validation with k=5 as model validation technique, are shown in Table 2. We use the Area under Roc curve (AUC) [7] as performance measure.

Table 3. Area under de curve ROC

Classifiers	*Credit-australian*	*Credit-german*	*Credit-approval*
1-NN	0.8110	**0.6855**	0.8110
C4.5	0.8530	0.6540	0.8530
RIPPER	**0.8605**	0.5985	**0.8605**
MLP	0.8390	0.6695	0.8390
SMO	0.8580	**0.6855**	0.8580
Naive Bayes	0.7595	0.6785	0.7595

Conclusions and future work

Nowadays there are a lot of datasets from the financial environment that are very important to the different automated decision-making systems, but these datasets have some characteristics that make this task more complicated. In this paper we compared six different classification techniques in credit environment: Nearest Neighbor, C4.5,

Repeated Incremental Pruning to produce Error Reduction, Multilayer Perceptron, and Sequential Minimal Optimization Algorithm for training a Support Vector classifier and Naive Bayes.

These techniques were compared by using the Area under the curve ROC due to the problem of the unbalanced classes present in these credit datasets. Our studies showed that SMO model turned out to be best classifier for Credit-Australian and Credit-approval dataset, but in the Credit-German dataset both 1-NN and SMO share the best performance.

Finally an analysis of statistical significance on classifiers that were compared, was not possible because the number of datasets was not enough to carry out this kind of analysis.

Acknowledgments

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación, and Centro de Innovación y Desarrollo Tecnológico en Cómputo), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

References

1. Li, Y., Zhou, Z.: Research on Model for Evaluating Risks of Venture Capital Projects. J. Risk Anal. Cris. Response. 1, 142–148 (2011).
2. Xiong, T., Wang, S., Mayers, A., Monga, E.: Personal bankruptcy prediction by mining credit card data. Expert Syst. Appl. 40, 665–676 (2013).
3. Petropoulos, A., Chatzis, S.P., Xanthopoulos, S.: A novel corporate credit rating system based on Student's-t hidden Markov models. Expert Syst. Appl. 53, 87–105 (2016).
4. Yang, Y., Gu, J., Zhou, Z.: Credit risk evaluation based on social media. Environ. Res. 148, 582–585 (2015).
5. Alickovic, E., Subasi, A.: Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier. J. Med. Syst. 40, 1–12 (2016).
6. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. Soft Comput. 19, 3369–3385 (2015).
7. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar, A. and Kang, B. (eds.) AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings. pp. 1015–1021. Springer Berlin Heidelberg, Berlin, Heidelberg (2006).
8. Lichman, M.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml.
9. Michie, E.D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning , Neural and Statistical Classification. (1994).

10. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13, 21–27 (1967).
11. Salzberg, S.L.: C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Mach. Learn. 16, 235–240 (1994).
12. Cohen, W.: Fast effective rule induction. Twelfth Int. Conf. Mach. Learn. 115–123 (1995).
13. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133 (1943).
14. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386–408 (1958).
15. Minsky, M.L., A. Papert, S.: Perceptrons. MIT Press (1969).
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by backpropagation error. Nature. 323, 533–536 (1986).
17. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. 1–21 (1998).
18. Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learn. 20, 273–297 (1995).
19. Duda, R., Hart, P., Stork, D.: Pattern Classification and Scene Analysis. (1973).
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. 11, 10–18 (2009).