

Telemarketing Success: Evaluation of Supervised Classifiers

Yosimar O. Serrano-Silva¹, Yenny Villuendas-Rey², Cornelio Yáñez-Márquez¹

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
CDMX, México

² Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo,
CDMX, México

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

Abstract. Nowadays telemarketing constitutes a way in which goods and services companies can access to possible potential customers through phone calls. Telemarketing campaigns are focused on offer to potential customers or users, contracting or buying a good or service. Ascertain a priori which phone calls will be successful is a competitive advantage to the companies due to this allow them to reduce costs and focus on most likely groups of potential customers which would contract or buy the goods or services offered. For this task it is necessary to classify the phone calls in successful and unsuccessful calls, which is possible using supervised classifier. In this paper, we tested some supervised classification algorithms and compare their performance, based on the Area under the ROC Curve, over different well-known telemarketing datasets.

Keywords: Telemarketing classification, supervised classification, unbalanced data.

1 Introduction

Nowadays telemarketing is an important strategy to increase the level of sales, looking for potential clients using different communication channels like phone calls, internet, etc. In fact financial institutions like banks and insurance companies are the most benefited of this kind of campaigns using different techniques like speech dictation systems [1] to checking incomplete sales where the telemarketer fails providing sales information to a certain client.

The main purpose of this kind of marketing is contacting a certain group of people to meet a specific goal (offer a service, insurance, credit card, etc.), but the problem of choosing the group of costumers willing to buy the service is considered NP-hard [2]. For this reason some approaches has been proposed to predict the success of telemarketing calls [3], where applying a decision support system using some techniques of data mining, automatically can predict the result of phone calls used to sell long term deposits.

In the literature we can find more examples of works about prediction in telemarketing environment using a different approach. Customer lifetime value is a variable considered as the value of a customer in terms of expected benefits based on future interactions with the customer and in [4] is used for predicting future behavior of that customers and in this way, improve the return-on investment.

Nevertheless, working with data sets from telemarketing environment has the disadvantage that, in most of the cases, have unbalanced classes and mixed attribute types [5], for which is very important to choose the classification models in accordance with this situation.

A situation like unbalanced classes is present in a dataset when one of the classes has more elements than the others. This situation represents a problem at the moment of work with this kind of datasets due to the fact that unbalanced classes, in general, creates biased learning. The consequences of this are reflected during the testing phase because the biased learning causes that the classifiers just recognize appropriately the elements of the ruling class and therefore, give us inaccuracy results.

In this article is presented an experimental work using different supervised classifiers with telemarketing datasets with the purpose to know which one has the best performance under the circumstances described above.

The rest of the paper is organized as follows. Section 2 details 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

One of the most common method applied to validate the classifier performance in multiple jobs within the literature is stratified cross-validation (SCV). This method is based on partitioning the data set into two complementary subsets. This couple of subsets is used for training and testing the classifier and the main purpose of this technique is keep the same class distribution in both subsets.

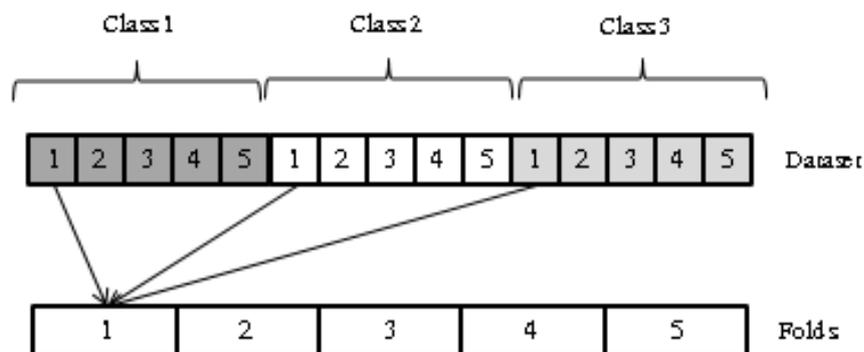


Fig. 1. Process to divide the dataset into k=5 subsets following the SCV technique.

To carry out this partition, as it shown in the Figure 1, it is necessary divide every class of the original dataset into k different partitions with the same number of patterns as possible. After that to form each fold, a partition of every class is taken.

Then one of these folds is use as testing data and the remaining k-1 folds are use as training data. Worth mentioning that this process is repeated k times and each fold is use as testing data exactly once.

The most popular value of k is 10, but in the case of unbalanced classes is most common use k equal to 5, in order to increase the performance of the algorithms, and to diminish the negative impact of unbalanced classes in the classification process. In addition, it is necessary to use a correct error measurement that can handle the problem of unbalanced classes and avoid inaccuracy results. The Area under the ROC curve (AUC) is a metric that comply with this requirement. This is a popular classification metric which exhibits the benefit of being independent of the class distribution (see Table 1):

Table 1. Confusion matrix.

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

$$\text{AUC} = (\text{TPR} + \text{TNR}) / 2, \quad (1)$$

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}), \quad (2)$$

$$\text{TNR} = \text{TN} / (\text{TN} + \text{FP}). \quad (3)$$

The results obtained with this measurement can be interpreted as ideal classification model if the value of the AUC is 1.0 and as random classifier if the value is 0.5. Moreover this measurement has been demonstrated that can be calculated as the average of the True negative Rate (TNR) and True Positive Rate (TPR) for discrete classifiers by Sokolova et al. [6].

This measurement has been used in different works e.g. to quantify the performance of imbalance learning ensembles [7] or to measure if there is a performance improvement as in [8] working with unbalanced classes.

3 Results and Discussion

3.1 Datasets

To accomplish the different experiments, four datasets that belong to telemarketing environments were used, with the characteristic of unbalanced classes. As you can see in Table 2 the unbalanced ratio is higher than 1.5 which means that in the four cases there is present the problem of unbalanced classes. This problem is important to consider because can produce a biased learning and inaccurate results using an inappropriate error measurement as mentioned earlier.

On the other hand, these datasets were taken from the Machine Learning repository of the University of California [9]. It is worth mentioning that the dataset were donated by S. Moro, P. Cortez and P. Rita [3] who obtained them from real bank data.

The Bank-full data corresponds to the original version of the dataset called Bank Marketing which has 45211 records with 17 attributes (7 numerical and 9 categorical) and do not have missing values. This dataset consist of information obtained from a direct marketing campaign of a Portuguese banking institution which was based on phones calls. Often, more than one contact was required to the same client in order to know if a bank term deposit would be or not subscribed. On the other hand, the Bank data is a sample of 10% from the Bank-full dataset randomly selected. This small dataset was made with the aim to test more computationally demanding machine learning algorithms.

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

<i>Data set</i>	<i>Instances</i>	<i>Attributes</i>	<i>Classes</i>	<i>Missing values</i>	<i>Unbalance Ratio</i>
Bank	4521	17	2	No	7.677
Bank-full	45211	17	2	No	7.548
Bank-additional	4119	21	2	No	7.956
Bank-additional-full	41188	21	2	No	7.876

The Bank-additional-full dataset is also based on the Bank marketing data, but this different version is enriched by the addition of new social and economic attributes. This dataset has 41188 instances with 21 attributes (10 numerical and 10 categorical). Also the Bank-additional dataset is a short version of the previous one, with 10% of the examples randomly selected from the Bank-additional-full data.

3.2 Algorithms to Compare

The following subsection provides a brief introduction of the most common classification models which were chosen for comparison of results in the next part of this work.

Nearest Neighbor (1-NN)

Nearest Neighbor model [10] follows the structure of a learning technique called instance-based learning. This classifier use a dissimilarity measure to carry out the classification of a pattern, using the closest instances of the training set, according to the measure selected, to give to each pattern from the testing set a class label. This model is based on the idea that every pattern from a dataset share some similar characteristics and properties with some individuals around.

The most popular similarity measure used when the dataset just has numerical attributes is the Euclidean distance (equation 4), and because of the use of a distance measure, this kind of model are called minimum distance classifiers.

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

C4.5

C4.5 is an algorithm that builds decision trees from a dataset based on information entropy concept [11]. This model chooses one of the attributes of the pattern that divide effectively its set of samples into better subsets. This process is repeated for each node and the criterion of splitting the subsets is the difference in entropy known as normalized information gain. In this way, the attribute with the highest normalized information gain value is selected to make the decision. By last this model has some base cases. The first case is when an instance of previously-unseen class is encountered, in this case the algorithm makes a decision node higher up the tree using the expected value. The second case is when none of the attributes provide any information gain. When this case occurs, once more the algorithm makes a decision node higher up the tree using the expected value. Finally, if the tree has made a decision node higher up the tree using the expected value, this model creates a leaf node for the decision tree indicating to choose the class.

Repeated Incremental Pruning to produce Error Reduction (RIPPER)

Repeated Incremental Pruning to produce Error Reduction is a classification model proposed in 1995 by William W Cohen [12]. This algorithm is based on the association rules with reduced error pruning (REP) and in fact is an optimized version of the IREP classifier.

In this kind of algorithms, the training data is split into other two sets: a growing set and a pruning set. Then a rule set is formed using some heuristic method to increase this set. This final rule set is simplified using one of the pruning operators and this would delete any single condition or any single rule; this process is repeated several times. At each stage of simplification, the preselected pruning operator is which returns the greatest reduction of error on the pruning set. This process ends when any pruning operator produces an increment in the error on the pruning set.

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.

Based on the work of McCulloch and Pitts, in 1957 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 an 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].

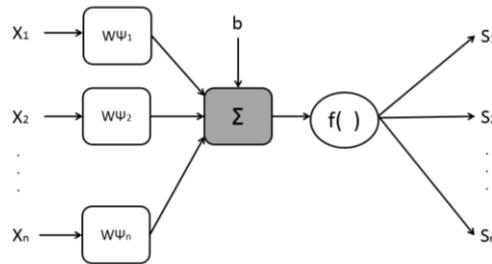


Fig. 2. Artificial neuron scheme.

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.

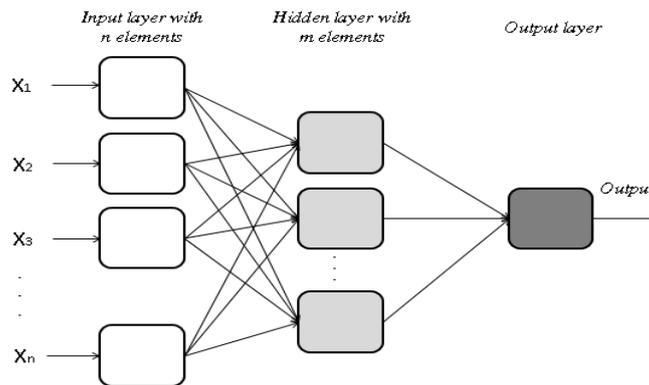


Fig. 3. General model of a MLP network with one hidden layer.

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 $(x_1, y_1), \dots, (x_n, y_n)$ 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:

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

subject to:

$$0 \leq \alpha_i \leq C, \quad \text{for } i = 1, 2, \dots, n, \quad (6)$$

$$\sum_{i=1}^n y_i \alpha_i = 0, \quad (7)$$

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:

$$0 \leq \alpha_1, \alpha_2 \leq C, \quad (8)$$

$$y_1 \alpha_1 + y_2 \alpha_2 = k. \quad (9)$$

And this reduced problem can be solved analytically. The algorithm proceeds as follows [18]:

- Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
- Pick a second multiplier α_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 $\{x_1, x_2, \dots, x_n\}$ 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:

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

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

$$\omega^* = \operatorname{argmax}_{\omega_j} p(\omega_i) \prod_{j=1}^n p(x_i | \omega_j). \quad (12)$$

Each was tested with the different datasets in Waikato Environment for Knowledge Analysis (WEKA) software [20] in its version number 3.6.13. The adequate parameter values for the algorithms were found by trial and error.

3.3 Discussion

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 3. We use the Area under Roc curve (AUC) [7] as performance measure.

Table 3. Area under the curve ROC.

<i>Classifiers</i>	<i>Bank</i>	<i>Bank-full</i>	<i>Bank-additional</i>	<i>Bank-additional-full</i>
1-NN	68.5500	64.5137	60.0203	64.5302
C4.5	65.9709	72.4634	69.6285	74.6412
RIPPER	69.1389	67.1002	76.6134	75.6068
MLP	67.4566	70.0040	66.6038	70.2377
SMO	57.3892	58.5780	63.8087	63.9084
Naive Bayes	70.7500	72.5500	75.6500	75.5500

As you can see in the Table 3, Naive Bayes was the model which obtain the best performance in two of the datasets and on the other hand, SMO was the worst algorithm in three of the four datasets. As well, the RIPPER model was the best working with the Bank-additional-full dataset and Bank-additional. Another aspect to highlight is that the distance used by the NN classifier could not have been the correct due to its results were not competitive as usual. The performance of the C4.5 is a special case because was very competitive in the full version of the datasets, but it was affected by the sampling of 10% of the other two datasets. Finally, it can be seen that the problem of unbalanced classes affect the performance of all the classifiers because, even when these are some of the most important models in the literature, they could not even reach an 80% of accuracy. In addition, it is worth mentioning that most of the classifiers got better results using the default parameters from WEKA, just in some cases like RIPPER or C4.5 there was an improvement modifying some values.

4 Conclusions and Future Work

In the telemarketing environment, there are some datasets that can be considered important to test automated decision-making systems, but in most of the cases, these datasets have some characteristics that make more complicated this task. In this work 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 algorithms were compared using the Area under the curve ROC due to the problem of the unbalanced classes present in telemarketing datasets. Our studies showed that Naïve Bayes Simple model turned out to be best classifier with the RIPPER model both getting the best performance in two datasets and on the other hand the SMO classifier got the worst performance in this comparative, even using different kernels.

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. Jung, D., Bae, M.-K., Choi, M.Y., Lee, E.C., Joung, J.: Speaker diarization method of telemarketer and client for improving speech dictation performance. *J. Supercomput.* 72, pp. 1757–1769 (2016)
2. Talla Nobibon, F., Leus, R., Spijksma, F.C.R.: Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. *Eur. J. Oper. Res.* 210, pp. 670–683 (2011)
3. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.* 62, pp. 22–31 (2014)
4. Moro, S., Cortez, P., Rita, P.: Using customer lifetime value and neural networks to improve the prediction of bank deposit subscription in telemarketing campaigns. *Neural Comput. Appl.* 26, pp. 131–139 (2014)
5. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. *Soft Comput.* 19, pp. 3369–3385 (2015)
6. 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)
7. Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C.I., Kuncheva, L.I.: Diversity techniques improve the performance of the best imbalance learning ensembles. *Inf. Sci. (Ny).* pp. 325, 98–117 (2015)
8. Frank, E., Bouckaert, R.R.: Naive Bayes for text classification with unbalanced classes. In: Furnkranz, J and Scheffer, T and Spiliopoulou, M. (ed.) *Knowledge discovery in databases: PKDD 2006, proceedings.* pp. 503–510. Springer-Verlag Berlin, Heidelberg Platz 3, D-14197 Berlin, Germany (2006)
9. Lichman, M.: *UCI Machine Learning Repository*, <http://archive.ics.uci.edu/ml>.
10. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. *Mach. Learn.* 6, pp. 37–66 (1991)
11. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers (1993).
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, pp. 115–133 (1943)
14. Rosenblatt, F.: *The perceptron, a perceiving and recognizing automaton* Project Para. Cornell Aeronautical Laboratory (1957)
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, pp. 533–536 (1986).
17. Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schölkopf, B., Burges, C.J.C., and Smola, A.J. (eds.) *Advances in Kernel Methods: Support Vector Learning* (1998)
18. Cortes, C., Vapnik, V.: Support-Vector Networks. *Mach. Learn.* 20, pp. 273–297 (1995)
19. Duda, R., Hart, P.: *Pattern Classification and Scene Analysis*. Willey, New York (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, pp. 10–18 (2009)