Expert Systems with Applications 39 (2012) 48–53

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Comparing alternative classifiers for database marketing: The case of imbalanced datasets

Ekrem Duman *, Yeliz Ekinci, Aydın Tanrıverdi

Dogus University, Industrial Engineering Department, Acibadem, 34722 Istanbul, Turkey

ARTICLE INFO

Keywords:
Database marketing
Imbalance datasets
Propensity modeling
Performance measures

ABSTRACT

There are various algorithms used for binary classification where the cases are classified into one of two non-overlapping classes. The area under the receiver operating characteristic (ROC) curve is the most widely used metric to evaluate the performance of alternative binary classifiers. In this study, for the application domains where the high degree of imbalance is the main characteristic and the identification of the minority class is more important, we show that hit rate based measures are more correct to assess model performances and that they should be measured on out of time samples. We also try to identify the optimum composition of the training set. Logistic regression, neural network and CHAID algorithms are implemented for a real marketing problem of a bank and the performances are compared.

1. Introduction

With the increase of the fierce competition in the last several decades, most companies are now aware of the importance of their existing customer base. Increasing the volume of relationship with the available customers is more profitable as compared to acquiring new customers which also helps in increasing customer loyalty. Such efforts of marketing to available customers are known as database marketing. Database marketing models aim at classifying consumers into buyers and non-buyers based on the predicted probabilities (Cui, Wong, Zhang, & Li, 2008). These models are also called binary classification models since they classify the consumers into two classes.

Database marketing models can be built for two different purposes: cross selling and up selling. In cross sell models the aim is to identify which customers are more likely to buy a particular product among the ones who do not have it. Whereas in up sell models, one tries to identify which customers may increase the volume on a particular product. In this study, we point out several critical points in increasing the success rates of both cross sell and up sell models. Our specific example will be a cross sell model developed for a bank.

As database marketers increasingly adopt innovative methods to provide decision support, assessing the performance of competing methods and model selection have become important issues. Due to the different performance criteria and validation procedures currently in practice, comparing various modeling methods is not always straightforward (Cui et al., 2008). The performance of the models is also affected from the balance of the classes in classification models.

The objectives of this study, particularly for database marketing, are to determine best composition (balance) of train set, to select the right performance measures to compare the models and to determine the data on which the comparisons should be carried out. For the training set composition we show that, imbalanced data sets where more examples of the minority class take place can perform better. The performance evaluation of classifiers should be determined by the application domain and the way the model results will be used. For a better comparison of models, they should be tested on unseen samples at a later period in time. Our main motivation in testing this strategy is the need and the way how these marketing models are used. Using the past data we develop models and use them in the future. So, their testing should be made on future data. We believe that our findings obtained from a real application in banking would contribute the related literature significantly.

In Section 2, the paper continues with the literature review. Section 3 gives a short description of the classification algorithms used in this study. Then in Section 4, the definitions of the performance criteria used for evaluation of the models are given. After presenting and discussing the results obtained in Section 5, summary and conclusions finalizes the paper in Section 6.

2. Literature review

In most binary classification problems as is also the case in our case study, usually the number of customers in one class is much more than the number of customers in the other class. This is

* Corresponding author.
E-mail addresses: eduman@dogus.edu.tr (E. Duman), yekinci@dogus.edu.tr (Y. Ekinci), atanriverdi@dogus.edu.tr (A. Tanrıverdi).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.06.048

known as the class imbalance problem (Japkowicz, 2000). When this is the case, a conventional machine learning algorithm usually has a poor classification performance for unseen samples from the minority class because it is strongly biased towards the majority class. Therefore, some approaches have been developed in the literature in order to overcome the class imbalance problems. The most popular approach is re-sampling, However, when faced with severe class imbalance (the number of customers in one class is much more than the number of customers in the other class; Japkowicz, 2000) and with a limited number of instances in the minority class, sampling methods become unreliable especially when the data distribution is unknown (Elazmeh, Japkowicz, & Matwin, 2006). Re-sampling can be done either by over-sampling the minority class or under-sampling the majority class (Liu, Hu, & Daren, 2008a). On the other hand, the problem of imbalance can also be coped for by assigning a higher misclassification cost when a positive instance is labeled incorrectly. Such options are available in the most popular data mining software packages like SAS and SPSS. Some researchers state that, after re-sampling, the number of examples from the two classes should be equal to each other for a better classification performance (Barandela, Sanchez, Garcia, & Rangel, 2003; Batista, Prati, & Monard, 2004; Efstathios, 2008; Liu, Hu, & Daren, 2008b). According to Baesens et al. (2003) better performance could be observed if the number of majority class samples is four times the number of minority class samples for credit risk models. In this study one of the questions that we seek for an answer is what should be the composition (or, balance) of the training set for a better learning of classification algorithms within the context of our case study (database marketing of a banking investment product). Japkowicz and Stephen (2002) indicate that the class imbalance problem is actually a relative problem that depends on the degree of class imbalance, the overall size of the training set and the classifier involved. If the overall size of the training set is large, the class imbalance problem may not be a handicap. Also the classifier used is an important factor, since some of the classification methods are believed to be less prone to the class imbalance problem such as the random forests. However, random forests are prone to overfitting also. This is more pronounced in noisy classification/regression tasks (Segal, 2004).

The second question that we try to answer in this study is how the performance of alternative models should be measured and compared. In the literature the most commonly used measures are accuracy based measures and the area under the receiver operating characteristic (ROC) curve (the AUC) (Bradley, 1997). The AUC shows the ability of a classifier to discriminate the two classes. Elazmeh et al. (2006) state that the only method to assess confidence of ROC curves is to construct ROC bands and in the case of severe class imbalance with few instances of the minority class, ROC bands become unreliable. In our opinion the performance evaluation of classifiers should be shaped according to the application domain and how the model results will be used. This is mainly due to the fact that in problem domains where there is a class imbalance, mostly the identification of minority class is more important than identifying the majority class. Consequently, a classifier which identifies the minority class better in the upper percentiles is more valuable. Actually this can equivalently be achieved if the models are compared only on the leftmost (say, one tenth) part of the ROC plane.

The third and last question this study raises is the environment (data) where the model performances should be measured. The typical practice followed in the literature is that after the samples from both classes are gathered together, some portion of it is used to train classifiers and the rest is used as a hold-out test set (Givargis & Karimi, 2009). This is quite logical since the algorithms tend to memorize the samples in the train set, a test on a separate set is more objective. In our study, we claim and show that for a more objective comparison of models, they should be tested on unseen samples at a later period in time.

In the following two sections some summary information about the algorithms and the performance criteria used in this study are given with the company of some additional literature review.

3. Algorithms for binary classification

There are many machine learning algorithms that we could have used and compared for our case study. Also in the software package we used, most of them are available with quite easy usage and in the past for some time we had tested all the alternative techniques. As a result of our experiences and paying attention to the results obtained by other researchers, in this study we decided to test and compare three different algorithms: the logistic regression, the neural networks and the Chi-squared automatic interaction detector (CHAID) algorithm.

Logistic regression is preferred as one of the algorithms to be compared since it is quite popular in risk prediction models (Baesens et al., 2003), financial classification modeling, and especially in direct marketing classification models. Neural network method is selected for classification since it was stated that it outperforms the other methods both for balanced and imbalanced datasets in previous studies (Baesens et al., 2003; Cui et al., 2008; Japkowicz & Stephen, 2002). CHAID is decided to be implemented in this study because of its proven performance in database marketing models among the other decision tree algorithms (Duman, 2006).

Their short descriptions are given below:

3.1. Logistic regression

Given a training set of N data points $D = \{(x_i, y_i)\}_{i=1}^{N}$, with input data $x_i \in R^n$ and corresponding binary class labels $y_i \in \{0, 1\}$, the logistic regression approach to classification tries to estimate the probability $P(y = 1|x)$ as follows:

$$p(y = 1|x) = \frac{1}{1 + \exp(-(w_0 + w^T x))}, \quad (1)$$

where $x \in R^n$ is an n-dimensional input vector, w is the parameter vector and the scalar w_0 is the intercept. The parameters w_0 and w are then typically estimated using the maximum likelihood procedure.

In logistic regression models dependent (predicted) variable is in categorical form, and has two or more levels. Independent variables may be in numerical or categorical form Camdeviren, Yazıcı, Akkus, and Sungur (2007).

3.2. Neural networks

A neural network is a parallel, distributed information processing structure consisting of processing elements interconnected together with unidirectional signal channels called connections. Each processing element has a single output connection which branches into as many collateral connections as desired (Hecht-Nielsen, 1988). Trained neural network finds the appropriate weights of inputs giving the closest value of the output. Fig. 1 provides an example of a neural network structure with one hidden layer and one output neuron where x_is are the inputs and y is the output. Training process allows finding the weights that generate the values in the hidden layer, using x_i, that are finally transformed to output values (y). Training a neural network model essentially means selecting one model from the set of allowed models that minimizes a predetermined criterion (Ekinci, Temur, Çelebi, & Bayraktar, 2010). There are numerous algorithms available for training neural

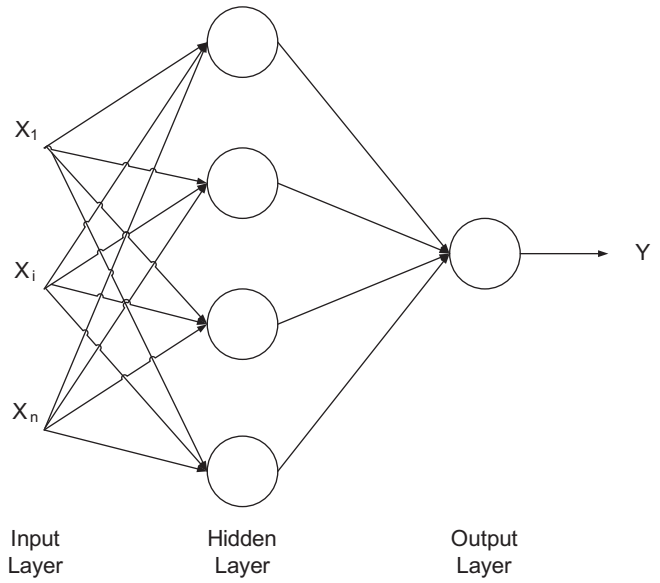

Fig. 1. A neural network structure with one hidden layer.

network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

3.3. CHAID

CHAID stands for chi-squared automatic interaction detector. It is a highly efficient statistical technique developed by Kass (1980). Using the significance of a statistical test as a criterion, it evaluates all of the values of a potential predictor field. It merges values that are judged to be statistically similar with respect to the target variable and maintains all other values that are dissimilar. It then selects the best predictor to form the first branch in the decision tree, such that each child node is made of a group of similar values of the selected field. This process continues recursively until the tree is fully grown. The statistical test used depends upon the measurement level of the target field. If the target field is continuous, an *F* test is used. If the target field is categorical, a chi-squared test is used.

Each of these three algorithm classes has some pros and cons. CHAID like all decision tree algorithms and the logistic regression can be trained quite fast whereas neural network training can take too much time especially for larger training sets. Neural network can learn with fewer data as compared to others and it is successful in identifying nonlinear relationships. As for the interpretation of model results, CHAID is the most preferable followed by the logistic regression and the neural network.

4. Performance criteria for binary classification

Binary classification is the most popular classification task where the input is to be classified into one, and only one, of two non-overlapping classes which are typically named as the positive and the negative classes (Sokolova & Lapalme, 2009). In this study, the two non-overlapping classes are buyers and non-buyers of a particular investment product of a bank.

The four counts, which constitute a confusion matrix (as seen in Table 1) for binary classification are: the number of correctly recognized positive class examples (true positives), the number of correctly recognized examples that belong to the negative class (true negatives), and examples that either were incorrectly assigned to the positive class (false positives) or that were not recognized as positive class examples (false negatives) (Sokolova & Lapalme, 2009).

Table 1
The confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Most often performance measures based on the values of the confusion matrix that are used to evaluate the performance of a classification model include hit rate, accuracy, capture rate, and lift (Sokolova & Lapalme, 2009). The performance criteria differ in their assumptions about the costs of misclassification errors and the types of errors that are used to measure the performance of classifiers (Cui et al., 2008).

Hit rate which is also known as precision, is the number of correctly classified positive examples divided by the number of examples labeled by the model as positive:

$$\text{Hit rate} = \text{TP}/(\text{TP} + \text{FP}). \tag{2}$$

Accuracy measures the overall effectiveness of a classifier:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FN} + \text{FP} + \text{TN}). \tag{3}$$

In a number of cases, accuracy may not be the most appropriate performance criterion since it tacitly assumes equal misclassification costs for false-positive and false-negative predictions (Baesens et al., 2003).

Capture rate also known as recall or sensitivity, is the number of correctly classified positive examples divided by the number of positive examples in the data. In other words it is the accuracy among the positive instances. This rate determines the effectiveness of a classifier to identify positive labels:

$$\text{Capture rate} = \text{TP}/(\text{TP} + \text{FN}). \tag{4}$$

Lift rate is the hit rate of a classifier in comparison with that identified by a random model or no model out of the total number of records at a given decile. In other words, for example if the ratio of buyers is 20% in the whole data and if the hit rate in a group of customers identified by the model is 60% then, the lift would be equal to three. It is seen as a degree of efficiency in marketing when it is made not randomly but based on model results. It is a measure well respected among marketers.

The ROC curve is a two-dimensional graphical illustration of the capture rate (or, true positive rate) on the *Y*-axis versus false positive rate on the *X*-axis for various values of the classification threshold (see Fig. 2). The area under this curve is called the *area under curve* (*AUC*). A model that perfectly discriminates between the TP and FP will have an area index equal to 1.0 and a model with no discriminatory power will result in an area index of 0.50. A model with a higher AUC value is said to outperform the alternatives or dominate the others (Cui et al., 2008). *Area under curve* (*AUC*) shows the classifier's ability to avoid false classification. AUC provides an estimate of the probability that a randomly chosen instance of class 1 (positive instance) is correctly rated (or ranked) higher than a randomly selected instance of class 0 (negative instance).

Other common performance measures are: *F*-measure (its evaluation focus is on relations between data's positive labels and those given by a classifier), Kappa statistic (it is originally a measure of agreement between two classifiers), mean absolute error (MAE shows how much the predictions deviate from the true probability), root mean squared error (it is just a quadratic version of MAE, which penalizes strong deviations from the true probability) and macro

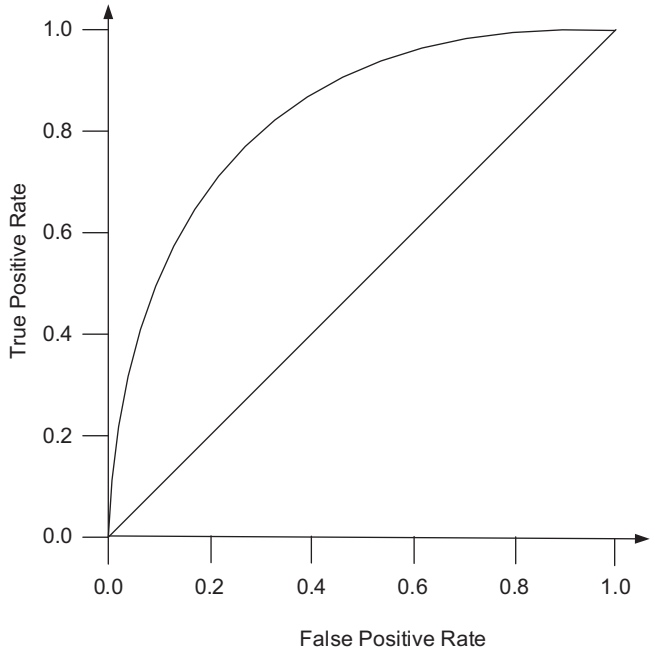

Fig. 2. A sample ROC curve.

average mean probability rate (MAPR is computed as an arithmetic average of the mean predictions for each class). Sokolova and Lapalme (2009) and Ferri, Hernández-Orallo, and Modroiu (2009) present a systematic analysis of twenty-four performance measures used in the complete spectrum of Machine Learning classification tasks, i.e., binary, multi-class, multi-labeled, and hierarchical.

In direct marketing applications, due to budget constraints and other considerations, typically only the names in the top deciles are targeted to send the promotion materials from a company (Zahavi & Levin, 1997). In other words, when the model scores showing the probability of positive class are sorted in a decreasing order, only the customers taking place at the top of the list are important or actionable in terms of marketing. Thus, the performance of a model over the entire data may not be relevant. In this study, the performance on top 1%, 5% and 10% of positive class scores (or, propensities to buy the product) are measured and compared.

5. Application in direct marketing

In our application we tried to determine the propensities of bank customers to buy a relatively unpopular investment product. For this purpose we were provided a sample data which included 2826 positive records (customers who bought the product) and 14130 negative records (customers who did not buy the product). This dataset was divided into two parts as training set and test set. Following the market practice, training set included 70% of the data and test set included 30% (Baesens et al., 2003). In order to find the optimum composition of the training set we tried six different imbalance figures that are obtained by deleting random instances of the appropriate class. Table 2 shows the number of positive and negative records existing in the training and test sets (in the table e.g. 1/6 means that the number of negative records is approximately six times the positive records).

Logistic regression (logit), neural network (NN) and CHAID algorithms are trained with their default settings in PASW modeler (version 13) on these six different training sets. We believed and also experienced that default settings are determined based on the experience of the SPSS Company and they perform rather good. In logistic regression algorithm, enter method was chosen (all model terms are entered in one step). The level of confidence interval for coefficients was taken as 95% and the likelihood ratio was used to test the significance of an independent variable in the model. Cut-off value 0.5 was chosen for classification. For the neural network algorithm, ten input neurons in one input layer together with one hidden layer which contained (after some internal comparisons) three neurons were used. The maximum tree depth was specified as five in CHAID algorithm. The Chi-Square statistic was calculated by Pearson method. For splitting and merging alpha values equaling to 0.05 were selected.

Table 2
Number of positive and negative records for different imbalance cases.

Balance	Number of positive records in training set	Number of negative records in training set	Number of positive records in test set	Number of negative records in test set
1/1	1929	2015	897	811
1/2	1929	3979	897	1673
1/3	1929	5962	897	2516
1/5	1929	9963	897	4167
1/6	1606	9958	748	4173
1/10	982	9902	431	4228

The model performance results for the whole test set and the top 1, 5 and 10 percentiles are given in Table 3. For the whole test set if the model score for positive class exceeds 50 it is assumed that the prediction is the positive class and based on this, hit rate and accuracy measures are calculated. In other words the cutoff point for the positive class is the score 50 or above (for NN we divided the score produced by two and added 0.5 to the quotion so that the result will have the same meaning as the results of the other two algorithms). Actually, the cut off value for the positive class prediction can also be taken as any value other than 50 and thus the value 50 here has only a psychological meaning and importance in that most people in marketing domain tend use it in their marketing applications. For example, in their *know your customer* screens they prefer to mark customers who has this score or above as having propensity to buy that product. Because of these reasons we wanted to investigate the model results in this respect.

In the top percentile columns, the cutoff point is determined as the lowest score in the named top percentile of the scores. This way the target list sizes will be equal to each other and a more meaningful comparison of the performances will be possible. Note that for a given number of positive predictions the performance measures hit rate, capture rate and lift are parallel and actually it is sufficient to tabulate only one of them but here for the convenience of the reader we tabulated all. In the table, the best values are highlighted in bold.

First of all we have to say that the performances of the three algorithms as tabulated in Table 3 are statistically indifferent at the level of alpha = 0.05 where we tested the equality of means by ANOVA for all columns of Table 3 (hit rate, accuracy and AUC). However, we still prefer to state the following statistically weak observations.

Let us look at the whole test set first. As mentioned earlier this is the place where most studies in the literature make the comparison of the algorithms. In terms of hit rates logit seems to be better than the others. However, as we look at the accuracy this is not the case and NN becomes slightly better than logit. The reason of this is that, logit predicts fewer numbers of customers in the positive class and thus its accuracy in that group of customers (hit rate) is better. In return since it labels some positive class customers as negative its overall accuracy degrades. Of course this is valid when the cutoff point is taken as 50. For a different cut off value

Table 3
Performance criteria results on the test data.

$B = 1/1$	Whole test set			Top 1%			Top 5%			Top 10%		
	Hit	Accuracy	AUC	Hit	Capture	Lift	Hit	Capture	Lift	Hit	Capture	Lift
Logit	**94.3**	75.7	0.836	**100**	**2**	**2**	**97.6**	**9.8**	**1.953**	95.3	19.1	1.906
NN	79.4	75	0.836	**100**	**2**	**2**	96.5	9.7	1.929	**96.5**	**19.3**	**1.929**
CHAID	79.3	**76.4**	**0.843**	94.1	1.9	1.882	92.9	9.3	1.859	92.9	18.6	1.859
$B = 1/2$												
Logit	**81.3**	79.5	0.839	**100**	**3.1**	**3**	**92.2**	**13.9**	**2.766**	90.6	27.2	2.718
NN	76.4	**79.6**	**0.839**	92.3	2.8	2.769	86.7	13.1	2.602	87.5	26.3	2.624
CHAID	81.2	79.3	0.838	96.2	2.9	2.885	86.7	13.1	2.602	**91.4**	**27.5**	**2.741**
$B = 1/3$												
Logit	**79.7**	**82.4**	**0.835**	**100**	**4.1**	**4**	**86.5**	**17.5**	**3.462**	80.9	32.5	3.238
NN	77.8	82.3	0.834	88.6	3.7	3.543	85.4	17.2	3.415	**83**	**33.4**	**3.32**
CHAID	72.5	82.3	0.834	85.7	3.5	3.429	85.4	17.2	3.415	80.1	32.2	3.202
$B = 1/5$												
Logit	**71.4**	86.9	**0.836**	**86.3**	**5.2**	**5.176**	**79.9**	**23.9**	**4.795**	**69.8**	**41.7**	**4.189**
NN	70.6	**87.1**	0.834	84.3	5.1	5.059	79.1	23.7	4.748	69.4	41.5	4.166
CHAID	69.4	86.7	0.835	84.3	5.1	5.059	73.6	22.1	4.417	69	41.3	4.142
$B = 1/6$												
Logit	**71.5**	87	**0.842**	**90**	**6.3**	**6.3**	70.4	24.4	4.931	62.1	43	4.345
NN	**71.5**	88.2	0.838	82	5.8	5.74	**76.1**	**26.4**	**5.328**	**62.9**	**43.5**	**4.402**
CHAID	62.8	**91.5**	0.841	74	5.3	5.18	73.3	26	5.13	62.3	43.1	4.359
$B = 1/10$												
Logit	59.9	90.9	**0.832**	**72.3**	**8**	**7.957**	59.8	33	6.581	**46.9**	**48.1**	**5.158**
NN	**61.5**	**91.6**	**0.832**	**72.3**	**8**	**7.957**	**62.4**	**34.4**	**6.863**	43	46.9	4.734
CHAID	**61.4**	87.2	0.829	53.2	5.9	5.851	53	29.2	5.829	43.7	47.4	4.805

the story can be different. However, for the business domains of high imbalance the behavior of logit can be preferable since the number of positive class cases is naturally fewer. As we go down the table hit rates gets smaller as the imbalance increases. This is natural since as the minority class becomes more minor, the algorithms face with difficulty in learning that class sufficiently. On the other hand, when we look at the famous AUC metric, we see that it is almost the same for all algorithms and balance figures with logit only slightly better than the others. As it is almost the same for all balances and since even the highest value is observed in $B = 1/1$, the conclusion of having equal number of positive and negative cases in training set (Barandela et al., 2003; Liu et al., 2008b) seems to be supported by our analysis also. However, the analysis on the top percentiles does not support this result.

Actually the analysis in the top percentiles of the test set is not very meaningful. This is because the figures are highly dependent on the imbalance value of the test set and in fact none of them is a good representative of the whole customer base where imbalance is much higher. Such an analysis on real life data is also made and will be discussed later. As we evaluate each balance figure separately what we observe is that logit outperforms the others especially in the top one and five percentiles. This is in parallel to the above explanation in that, as logit produces fewer number of positive class predictions it has more accuracy in the upper percentiles. Note that for higher imbalances, the hit rates in top 10 percentile are lower than the hit rates in the whole test data set since the number of customers getting a score of 50 or more becomes less than 10% of all customers.

Next, we evaluated the performances of the algorithms on the real purchase data in a later period of time. This data contained 412 product buyers and 169,777 non-buyers. The sales ratio is only about 0.2%. We applied the models trained above on all these 170,189 customers which corresponded to the full population. Note that when we are talking about the full population we do not need any statistical tests and any algorithm performing better than the others (even if, slightly) should be respected as the best algorithm.

After applying the models to the full population we determined the customers' propensities to buy our product and sorted them with a decreasing manner. Then, we recorded the hit rates on top 1% (1702 customers), top 5% (8509 customers) and top 10% (17,019 customers). Obviously, the marketing department will have a limit on the budget of their campaign and thus they will target a campaign to a limited number of customers. Through our conversations with marketers we learnt that at most 10 times the natural buyers (which make 4120 customers in our case) can be a good limit on the size of the campaign list. Accordingly, we recorded the hit rates on top 4120 customers also. Note that this corresponds to top 2.4%, a figure between top 1% and top 5%. The results are given in Table 4 where this time we tabulated only the hit rate which is a representative of capture rate and lift figures also.

When compared to previous observations derived from Table 3, Table 4 brings some surprises. While in the top 1% NN is slightly better, in the other percentiles and in the top 4120 customers, CHAID outperforms the others. Surprisingly logit, the winner in Table 3, turned out to be very poor here. As the real use of these marketing models will be like here, and as we are talking about the full population, the results of Table 4 are more prominent. Thus, we can conclude that, as opposed to what is done in many works in the literature it is not correct to compare the performance of alternative models and pick the best one based on a hold-out test set.

Since the natural imbalance of the whole population is quite larger than the imbalance values of any of our training sets, the hit rates observed in different imbalance blocks of Table 4 are comparable (this was not the case in Table 3). Also, independent from the underlying imbalance in the training set, a classifier which has a higher hit rate for an imposed campaign list size is obviously better than the others. For example, if a larger campaign list is preferred (top 10% of customers), a much more successful result can be obtained with CHAID when it is trained on a $B = 1/6$ training set. For a more general conclusion where a campaign list size is not specified, we observe that the $B = 1/3$ and $B = 1/5$ cases have higher hit rates than the others. This observation does not support the previous studies (Barandela et al., 2003; Batista et al., 2004; Efstathios, 2008; Liu et al., 2008b) and once again it contradicts with the results of Table 3.

Table 4
Performances on future data.

Hit rates	Top 1%	Top 5%	Top10%	Top 4120
$B = 1/1$				
Logit	4	2	1.2	3
NN	**5.3**	2.1	1.4	**3.1**
CHAID	2.7	**2.2**	**1.4**	3
$B = 1/2$				
Logit	3.9	1.9	1.2	2.9
NN	**4.8**	**2.2**	1.4	3
CHAID	3.3	**2.2**	**1.5**	**3.2**
$B = 1/3$				
Logit	3.9	1.9	1.7	3
NN	**5.3**	**2.2**	1.4	3
CHAID	4.1	**2.2**	**1.9**	**3.2**
$B = 1/5$				
Logit	3.8	1.9	**1.2**	2.9
NN	4.3	**2.2**	1.4	3.1
CHAID	**4.8**	**2.2**	1.3	**3.2**
$B = 1/6$				
Logit	4.2	1.9	3.2	2.9
NN	**4.7**	**2.2**	4.4	**3.1**
CHAID	3.1	**2.2**	**5.3**	**3.1**
$B = 1/10$				
Logit	4	1.9	1.2	3
NN	**4.8**	**2.2**	1.4	3.3
CHAID	**4.8**	2.1	**1.4**	3

6. Summary and conclusions

In this study the problem of developing successful database marketing models is taken up based on a specific example from banking industry. Three different data mining algorithms namely the logistic regression, the neural network and CHAID are implemented and compared. However, the main emphasis of the paper is not comparing the performance of particulary these algorithms but rather how different algorithms or models should be compared to each other in order to find the most useful one to implement for real life use. The study produced some very important results which we itemize and list below:

(i) The hit rate or the related capture rate or lift metrics are more meaningful than the popular AUC metric.
(ii) The performance of models should be compared on an unseen data from a later period in time and not on a hold-out test sample separated from the sample which could also be used as a training data.
(iii) It is good to have more examples of negative class in the training set.
(iv) CHAID is a successful algorithm for developing database marketing models.

The above conclusions are driven for the database marketing domain. But, obviously they can be generalized for all domains where there is a high degree of imbalance in the data and it is more important to identify positive cases than the major negative cases. As a future work the comparisons and analyses made here can be extended to include more algorithms, more products from banking or other problem domains to draw a more general picture for the principles of *model comparison*.

References

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*, 627–635.

Barandela, R., Sanchez, J. S., Garcia, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition, 36*, 849–851.

Batista, G., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations, 6*(1), 20–29.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.

Camdeviren, H. A., Yazıcı, A. C., Akkus, Z., & Sungur, M. A. (2007). Comparison of logistic regression model and classification tree: An application to postpartum depression data. *Expert Systems with Applications, 32*, 987–994.

Cui, G., Wong, M. L., Zhang, G., & Li, L. (2008). Model selection for direct marketing: Performance criteria and validation methods. *Marketing Intelligence & Planning, 26*(3), 75–292.

Duman, E. (2006). Comparison of decision tree algorithms in identifying bank customers who are likely to buy credit cards. In *Proceedings of the workshop on information technologies for business, seventh international baltic conference on databases and information systems, 3–6 July*, Kaunas, Lithuania.

Efstathios, S. (2008). Author identification: Using text sampling to handle the class imbalance problem. *Information Processing and Management, 44*, 790–799.

Ekinci, Y., Temur, G. T., Çelebi, D., & Bayraktar, D. (2010). Company success estimation during economic crisis: An artificial neural network based approach. *Endüstri Mühendisliği Dergisi, 21*(1), 17–29.

Elazmeh, W., Japkowicz, N., & Matwin, S. (2006). Evaluating misclassifications in imbalanced data. *Lecture Notes in Computer Science, 4212*, 126–137.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters, 30*, 27–38.

Givargis, Sh., & Karimi, H. (2009). Mathematical, statistical and neural models capable of predicting LA, max for the Tehran–Karaj express train. *Applied Acoustics, 70*, 1015–1120.

Hecht-Nielsen, R. (1988). Theory of the backpropagation neural network. In *Proceedings of the international joint conference on neural networks* (pp. 1–593).

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 international conference on artificial intelligence*, *IC-AI'2000* (pp. 111–117).

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

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of Applied Statistics, 29*(2), 119–127.

Liu, J., Hu, Q., & Daren, Y. (2008a). A weighted rough set based method developed for class imbalance learning. *Information Sciences, 178*, 1235–1256.

Liu, J., Hu, Q., & Daren, Y. (2008b). A comparative study on rough set based class imbalance learning. *Knowledge-Based Systems, 21*, 753–763.

Segal, M. R. (2004). *Machine learning benchmarks and random forest regression*. UC San Francisco: Center for Bioinformatics and Molecular Biostatistics<http://escholarship.org/uc/item/35x3v9t4>. Retrieved from:.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45*, 427–437.

Zahavi, J., & Levin, N. (1997). Applying neural computing to target marketing. *Journal of Direct Marketing, 11*(4), 76–93.