APPLICATION OF DATA MINING TECHNIQUES FOR DIRECT MARKETING

Anatoli Nachev

***Abstract**: This paper presents a case study of data mining modeling techniques for direct marketing. It focuses to three stages of the CRISP-DM process for data mining projects: data preparation, modeling, and evaluation. We address some gaps in previous studies, namely: selection of model hyper-parameters and controlling the problem of under-fitting and over-fitting; dealing with randomness and 'lucky' set composition; the role of variable selection and data saturation. In order to avoid overestimation of the model performance, we applied a double-testing procedure, which combines cross-validation, multiple runs over random selection of the folds and hyper-parameters, and multiple runs over random selection of partitions. The paper compares modeling techniques, such as neural nets, logistic regression, naive Bayes, linear and quadratic discriminant analysis, all tested at different levels of data saturation. To illustrate the issues discussed, we built predictive models, which outperform those proposed by other studies.*

***Keywords**: direct marketing, data mining, modelling, classification, variable selection, neural networks.*

***ACM Classification Keywords**: I.5.2- Computing Methodologies - Pattern Recognition – Design Methodology - Classifier design and evaluation.*

Introduction

Today, banks are faced with various challenges offering products and service to their customers, such as increasing competition, continually rising marketing costs, decreased response rates, at the same time not having a direct relationship with their customers. In order to address these problems, banks aim to select those customers who are most likely to be potential buyers of the new product or service and make a direct relationship with them. In simple words, banks want to select the customers who should be contacted in the next marketing campaigns.

Response modeling is usually formulated as a binary classification problem. The customers are divided into two classes, respondents and non-respondents. Various classification methods (classifiers) have been used for response modeling such as statistical and machine learning methods. They use historical purchase data to train and then identify customers who are likely to respond by purchasing a product.

Many data mining and machine learning techniques have been involved to build decision support models capable of predicting the likelihood if a customer will respond to the offering or not. These models can perform well or not that well depending on many factors, an important of which is how training of the model has been planned and executed. Recently, neural networks have been studied in [Elsalamon & Elsayad, 2013], [Elsalamony, 2014], [Moro et al., 2011], [Yu & Cho, 2006] and regarded as an efficient modelling technique. Decision trees have been explored in [Elsalamon & Elsayad, 2013], [Elsalamony, 2014], [Moro et al., 2011], [Sing'oei & Wang, 2013]. Support vector machines are also well performing models discussed in [Moro et al., 2011], [Shin & Cho, 2006], [Yu & Cho, 2006]. Many other modelling techniques and approaches, both statistical and machine learning, have been studied and used in the domain.

In this paper, we explore five modeling techniques and discuss factors, which affect their performance and capabilities to predict. We extend the methodology used in [Elsalamon & Elsayad, 2013], [Elsalamony, 2014], [Moro et al., 2011] addressing certain gaps.

The remainder of the paper is organized as follows: section 2 provides an overview of the data mining techniques used; section 3 discusses the dataset used in the study, its features, and the preprocessing steps needed to prepare the data for experiments; section 4 presents and discuses the experimental results; and section 5 gives the conclusions.

Data Mining Techniques

We often suspect some relationships among the data we wish to process. However, in order to make more precise statements, draw conclusions, or predict from the measured data, we have to set up a model which represents the nature of the underlying relationship. Here we use several modeling technique, namely: neural networks, logistic regression, naïve Bayes, and linear / quadratic discriminant analysis. This section briefly outlines each.

Neural Networks

A variety of neural network models are used by practitioners and researchers for clustering and classification, ranging from very general architectures applicable to most of the learning problems, to highly specialized networks that address specific problems. Among the models, the most common is the multilayer perceptron (MLP), which has a feed-forward topology. Typically, an MLP consists of a set of input nodes that constitute the input layer, an output layer, and one or more layers sandwiched between them, called hidden layers. Nodes between subsequent layers are fully connected by weighted connections so that each signal travelling along a link is multiplied by its weight w_{ij}. Hidden and output nodes receive an extra bias signal with value 1 and weight θ. The input layer, being the first layer, has size (number of nodes), which corresponds to the size of the input samples. Each hidden and output node computes its activation level by s_i (1) and then transforms it to output by an activation function $f_i(s_i)$. The NN we use in this study works with the logistic activation function (1), where β is slope parameter.

$$s_i = \sum_j w_{ij} x_j + \theta \qquad f_i(s_i) = \frac{1}{1+e^{-\beta s_i}} \tag{1}$$

The overall NN model is given in the form:

$$y_i = f_i(w_{i,\theta} + \sum_{j=I+1}^{I+H} f_j(\sum_{n=1}^{I} x_n w_{m,n} + w_{m,\theta}) w_{i,n}) \tag{2}$$

where y_i is the output of the network for node i, $w_{i,j}$ is the weight of the connection from node j to i and f_j is the activation function for node j. For a binary classification, there is one output neuron with logistic activation function. The training algorithm we use for the MLP is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [Broyden , 1970], [Fletcher , 1970]. The BFGS method approximates the Newton's method, a class of hill-climbing optimization techniques. The algorithm stops when the error slope approaches zero or after a maximum of epochs.

Logistic Regression

Logistic regression extends the ideas of linear regression. As with multiple linear regression, the independent variables $x_1, \ldots, x_q$ may be categorical or continuous variables or a mixture of these two types, but the dependent output variable Y is categorical, as we use logistic regression for classification. The idea behind logistic regression is straightforward: instead of using Y as the dependent variable, we use a function of it, which is called the *logit*. To understand the logit, we take two intermediate steps: First, we look at P, the probability of belonging to class 1, P can take any value in the interval [0, 1]. However, if we express P as

a linear function, it is not guaranteed that the right hand side will lead to values within the interval [0, 1]. The fix is to use a non-linear function of the predictors in the form:

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q)}} \tag{3}$$

This is called the logistic response function. For any values of $x_1, \ldots, x_q$ the right hand side will always lead to values in the interval [0, 1].

The next step is to use a cutoff value on these probabilities in order to map each case to one of the class labels. For example, in a binary case, a cutoff of 0.5 means that cases with an estimated probability of *P(Y = 1) > 0.5* are classified as belonging to class 1, whereas cases with *P(Y = 1) < 0.5* are classified as belonging to class 0. This cutoff need not be set at 0.5.

Naive Bayes

Bayesian classifiers [Clark & Niblett, 1989] operate by using the Bayes theorem, saying that: Let *X* be the data record (case) whose class label is unknown. Let *H* be some hypothesis, such as "data record *X* belongs to a specified class *C*." For classification, we want to determine *P(H|X)* - the probability that the hypothesis *H* holds, given the observed data record *X*. *P(H|X)* is the posterior probability of *H* conditioned on *X*. Similarly, *P(X|H)* is posterior probability of *X* conditioned on *H*. *P(X)* is the prior probability of *X*. Bayes theorem is useful in that it provides a way of calculating the posterior probability, *P(H|X)*, from *P(H), P(X)*, and *P(X|H)*. Bayes theorem is

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)} \tag{4}$$

A difficulty arises when we have more than a few variables and classes - we would require an enormous number of observations (records) to estimate these probabilities. Naive Bayes classification gets around this problem by not requiring that we have lots of observations for each possible combination of the variables. In other words, Naive Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. This assumption is called class conditional independence. It is made to simplify the computation and in this sense considered to be Naive. This assumption is a fairly strong assumption and is often not applicable, but bias in estimating probabilities often may not make a difference in practice - it is the order of the probabilities, not their exact values, which determine the classifications. Studies comparing classification algorithms have found the Naive Bayesian classifier to be comparable in performance with classification trees and with neural network classifiers. They have also exhibited high accuracy and speed when applied to large databases.

Linear and Quadratic Discriminant Analysis

Discriminant analysis [Fisher, 1936] is a technique for classifying a set of observations into predefined classes. Based on the training set, the technique constructs a set of functions of the predictors, known as discriminant functions. In principle, any mathematical function may be used as a discriminating function. In case of the linear discriminant analysis (LDA), a linear function (5) is used, where x_i are variables describing the data set.

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \tag{5}$$

The parameters a_i have to be determined in such a way that the discrimination between the groups is best, which means that the separation (distance) between the groups is maximized, and the distance within the groups is minimized. Quadratic discriminant analysis (QDA) is a generalization of LDA. Both LDA and QDA assume that the observations come from a multivariate normal distribution. LDA assumes that the groups

have equal covariance matrices. QDA removes this assumption, allowing the groups to have different covariance matrices. LDA is simpler, faster, and more accurate than QDA, but performing well mainly with linear problems - that is where the decision boundaries are linear. In contrast, QDA is suitable for problems with non-linear decision boundaries.

Dataset and Preprocessing

The direct marketing dataset used in this study was provided by Moro et al. [Moro et al., 2011], also available in [Bache, & Lichman, 2013]. It consists of 45,211 samples, each having 17 attributes (see Table 1), one of which, *y*, s the class label. The attributes are both categorical and numeric and can be grouped as:

- demographical (*age*, *education*, *job*, *marital status*);
- bank information (*balance*, *prior defaults*, *loans*);
- direct marketing campaign information (*contact type*, *duration*, *days* since last contact, *outcome* of the prior campaign for that client, etc.)

Table 1: Dataset attribute names, types, descriptions, and values.

#	Name (type)	Description: values
1	age (numeric)	
2	job (categorical)	type of job: "blue-collar", "admin.", "student", "unknown", "unemployed", "services", "management", "retired", "housemaid", "entrepreneur", "self-employed", "technician"
3	marital (categorical)	marital status: "married", "divorced", "single"
4	education (categorical)	"unknown", "secondary", "primary", "tertiary"
5	default (binary)	has credit in default? "yes", "no"
6	balance (numeric)	average yearly balance, in euros
7	housing (binary)	has housing loan? "yes", "no"
8	loan (binary)	has personal loan? "yes", "no"
9	contact (categorical)	contact communication type: "unknown", "telephone", "cellular"
10	day (numeric)	last contact day of the month
11	month (categorical)	last contact month of year: "jan", "feb", "mar", ..., "dec"
12	duration (numeric)	last contact duration, in seconds
13	campaign (numeric)	number of contacts performed during this campaign and for this client
14	pdays (numeric)	number of days that passed by after the client was last contacted from a previous campaign
15	previous (numeric)	number of contacts performed before this campaign and for this client
16	poutcome (categorical)	outcome of the previous marketing campaign: "unknown", "other", "failure", "success"
17	y (binary)	output variable (desired target): "yes", "no"

There are no missing values. Further details about data collection, understanding, and initial preprocessing steps can be found in [Moro et al., 2011]. Referring to the data understanding stage of the CRISP-DM, we explored each variable distribution using histograms, but in order to understand value distributions in the details, we did marginal distribution plots. They show relationship between any two variables examining the distribution of one, using the other one for grouping. Figure 1 illustrates marginal distribution plots of all 17 variables, using the class variable for grouping. Each variable distribution is represented by two histograms - one belonging to the class label 'no' and one to 'yes'. The last plot corresponding to the output variable *y* shows that the dataset is unbalanced. Indeed, the successful samples corresponding to the class 'yes' are 5,289, which is 11.7% of all samples; all other samples belong to the 'no' class, which is 88.3% of the dataset.

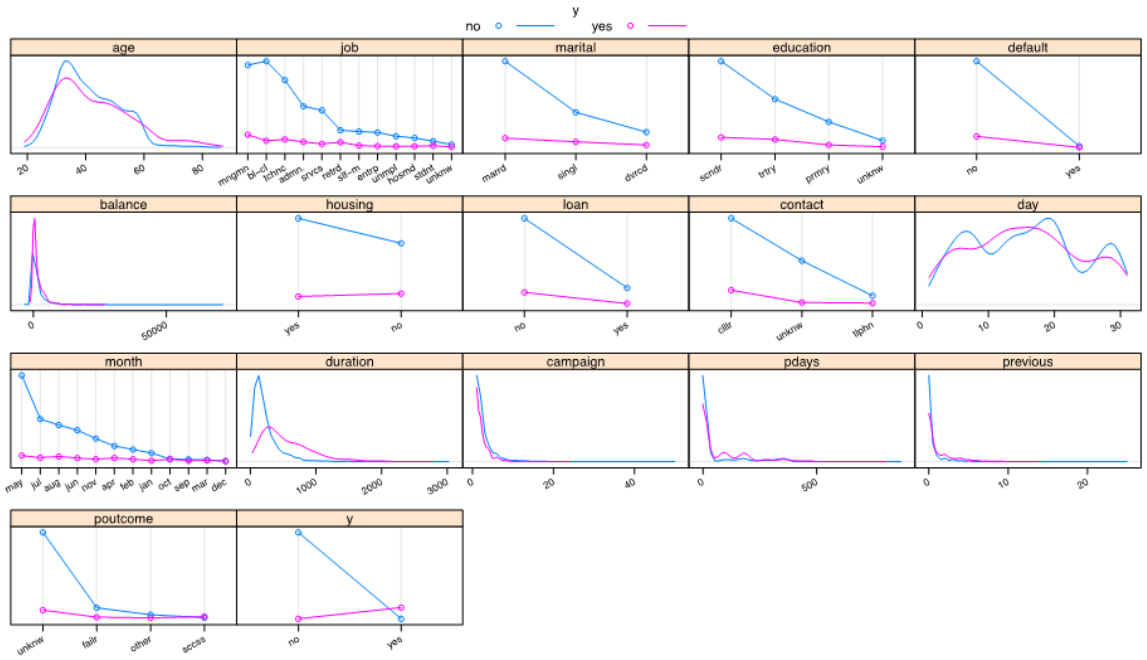

Fig. 1. Marginal distribution plots of all 17 variables, using the class variable y for grouping. Each variable distribution is represented by two histograms - one belonging to the class 'no' and one to 'yes'.

Some modelling techniques, like neural nets, process numeric data only in a fairly limited range, usually [0,1]. This presents a problem, as the dataset we use contains both numeric values out of the usual range and non-numeric. The data transformations needed to sort that out are part of the data preparation stage of the CRISP-DM project model [Chapman et al., 2000]. We did two transformations: mapping non-numeric data to binary dummies and normalization/scaling into the [0,1] interval.

Non-numeric categorical variables cannot be used as they are and must be decomposed into a series of dummy binary variables. For example, a single variable, such as *education* having possible values of "unknown", "primary", "secondary", and "tertiary" would be decomposed into four separate variables: *unknown* - 0/1; *primary* - 0/1; *secondary* - 0/1; and *tertiary* - 0/1. This is a full set of dummy variables, which number corresponds to the number of possible values. In this example, however, only three of the dummy variables need - if the values of three are known, the fourth is also known. Thus, we can map a categorical variable into dummies, which are one less than the number of possible values. Using reduced number of dummies we converted the original dataset variables into 42 numeric variables altogether, which is 6 less than the 48 variables used in [Elsalamon & Elsayad, 2013] and [Elsalamony, 2014]. There are two benefits of that: first, the neural network architecture becomes simpler and faster; secondly, in some modeling algorithms, such as multiple linear regression or logistic regression, the full set of dummy variables will cause the algorithm to fail due to the redundancy.

The second data transformation we did is related to normalization/scaling. This procedure attempts to give all data attributes equal weight, regardless of the different nature of data and/or different measurement units, e.g. *day* (1-31) vs. *duration* in seconds (0-4918). If the data are left as they are, the training process is getting influenced and biased by some 'dominating' variables with large values. In order to address this, we did normalization (z-scoring) and scaling down to [0, 1] by (6):

$$x_i^{new} = \frac{x_i - \mu}{\sigma}, \qquad x_i^{new} = \frac{x_i - a}{b - a} \tag{6}$$

where μ is the mean and σ is the standard deviation of the variable in question; [a,b] is the range of values for that variable. The two transformations were applied to each of the variables independently and separately.

Another part of the data understanding stage of CRISP-DM is to measure correlations between all variables of the dataset. We did pairwise correlation analysis to determine the extent to which values of pairs of variables are proportional to each other, as this may influence the model performance. Correlation coefficients can range in [-1,+1], where -1 represents a perfect negative correlation while +1 represents a perfect positive correlation. A value of 0 represents a lack of correlation. We used Spearman's rho statistic to estimate a rank-based measure of association. This method is robust and recommended if the data do not necessarily come from a bivariate normal distribution. After doing complete analysis of all 42 variables, we further focused to only 11 of them, those with significant correlation coefficients out of the interval [-0.4,+0.4]. Figure 2 shows the correlation coefficients in tabular format and by plot where ellipses converging to circles represent no correlations; the level of stretching of the ellipses shows the level of correlation. Colors and shades also illustrate whether correlations are positive or negative.

	education.secondary	education.primary	job.retired	month.may	housing	marital.married	poutcome.other	marital.divorced	poutcome.failure	pdays	previous
education.secondary	1.00	-0.43	-0.04	0.08	0.10	-0.02	0.01	0.02	0.00	0.00	0.00
education.primary	-0.43	1.00	0.13	0.03	0.01	0.14	-0.01	-0.01	-0.02	-0.03	-0.04
job.retired	-0.04	0.13	1.00	-0.07	-0.16	0.08	0.00	0.05	-0.01	0.01	0.02
month.may	0.08	0.03	-0.07	1.00	0.43	-0.04	0.01	0.01	0.03	0.02	0.00
housing	0.10	0.01	-0.16	0.43	1.00	0.01	0.04	0.00	0.11	0.08	0.07
marital.married	-0.02	0.14	0.08	-0.04	0.01	1.00	-0.03	-0.44	-0.01	-0.03	-0.03
poutcome.other	0.01	-0.01	0.00	0.01	0.04	-0.03	1.00	0.01	-0.07	0.43	0.44
marital.divorced	0.02	-0.01	0.05	0.01	0.00	-0.44	0.01	1.00	0.00	0.00	0.00
poutcome.failure	0.00	-0.02	-0.01	0.03	0.11	-0.01	-0.07	0.00	1.00	0.75	0.73
pdays	0.00	-0.03	0.01	0.02	0.08	-0.03	0.43	0.00	0.75	1.00	0.99
previous	0.00	-0.04	0.02	0.00	0.07	-0.03	0.44	0.00	0.73	0.99	1.00
poutcome.success	-0.02	-0.03	0.06	-0.06	-0.09	-0.02	-0.04	-0.01	-0.07	0.36	0.39
education.tertiary	-0.67	-0.27	-0.07	-0.12	-0.09	-0.09	0.00	-0.01	0.01	0.02	0.03
job.management	-0.40	-0.17	-0.12	-0.08	-0.06	-0.04	0.00	0.01	0.01	0.01	0.02

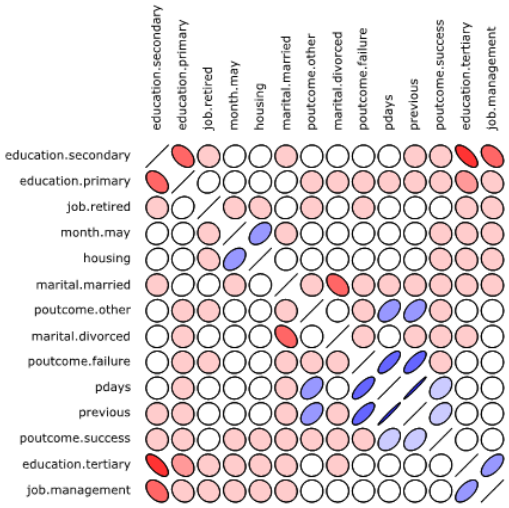

Fig. 2. Selection of most correlated variables (with absulute correlation coefficints above 0.4) and displayed tabularly and by a plot. Circles reprsent no correlation; ellipses represenit different level of correlation. Colors and shades represent how positive or negative correlations are

This analysis led us to the conclusion that the variables *pdays* and *previous* are in strong correlation, both related to past contacts with the client. There are also correlations between these two variables and the *poutcome* variable, which measures the outcome form previous campaigns. These variables can be identified as candidates for elimination at the modeling stage. We further did sensitivity analysis in order to measure the discriminatory power of each variable and its contribution to the classification task. We found that the least significant variable is *loan*, preceded by *marital*. On the other end, the most significant variables are *duration* and *month*.

Empirical Results and Discussion

In order to build models for direct marketing application and compare their characteristics with those discussed in other studies [Moro et al., 2011], [Elsalamon & Elsayad, 2013], [Elsalamony, 2014], we used the same dataset as used before and did experiments consistently. We, however, extended the methodology addressing the following issues:

- *Optimization of neural network architecture.* Simplifying the NN architectures may lead to better performance, easier to train, and faster NNs.
- *Validation and testing.* Using validation and test sets in a double-testing procedure helps to avoid overestimation of the model performance which is a gap in the studies mentioned above.
- *Randomness and 'lucky' set composition*. Random sampling is a fair way to select training and testing sets, but some 'lucky' draws can train the model much better than others. Using rigorous testing and validation procedures can solidify the conclusions made.
- *Variable selection*. Further to identifying importance of variables and their contribution to the classification task, also reported in the previous studies, we take the next step considering elimination of some input variables, which may lead to improvement of the model performance.
- *Data saturation*. We also explored the capacity of the modelling techniques to act in early stages of data collection where lack of sufficient data may lead to underfitting of the models.

All experiments were conducted using R environment [Cortez, 2010], [R Development Core Team, 2009], [Sing et al., 2005]. The model performances were measured by accuracy as the most common figure of merit, but on the other hand, accuracy can vary dramatically depending on class prevalence, thus being misleading estimator in cases where the most important class is underrepresented - that is our case, because the dataset is unbalanced with underrepresented class 'yes'. In order to address this problem, we used sensitivity, specificity, and ROC analysis [Fawcett, 2005] as more relevant performance estimators.

For the sake of consistency with the previous studies, we used 98 % of the dataset for training and validation, which was split randomly in ratio 2/3: 1/3. The rest of 2% were retained for test. Search of optimal NN architecture was made exploring models with one hidden layer of size from 0 to 13. In order to validate the results and reduce the effect of lucky set composition, each architecture was tested 300 times: internally, the fit algorithm runs 10 times with different random selection of training and validation sets and initial weights. For each of those set compositions, the 3-fold cross-validation creates 3 model instances and averages their results. We iterated all those procedures 10 times per architecture, recording and averaging accuracy and AUC.

H	*ACC*	*ACCmax*	*AUC*	*AUCmax*
0	89.514	92.040	0.895	0.937
1	89.011	92.150	0.898	0.910
2	89.849	93.143	0.903	0.924
3	90.489	93.143	0.906	0.939
4	90.289	91.107	0.908	0.913
5	90.250	90.606	0.910	0.921
6	90.090	90.701	0.912	0.919
7	90.285	90.606	0.913	0.919
8	90.025	90.700	0.915	0.923
9	90.049	90.505	0.915	0.922
10	90.050	90.505	0.915	0.920
11	90.091	90.800	0.913	0.918
12	89.528	90.300	0.913	0.923
13	90.120	90.403	0.914	0.918

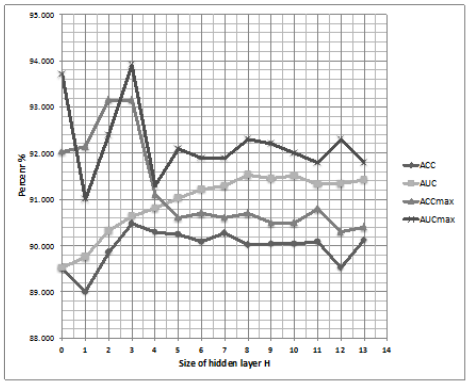

Fig. 3. Validated performance metrics of neural networks with H hidden nodes and architecture 42-H-1, where accuracy (ACC) and area under ROC curve (AUC) values are average of 300 model instances of that architecture. ACC_{max} and AUC_{max} are maximal values obtained.

We also explored how variable selection affects the models performance. Applying backward selection method based on variable significance, we found that eliminating the least important variable *loan* improves

the overall model performance. Figure 3 outlines results. NN models with 41-3-1 architecture show best average accuracy of 90.489%. They outperform the 48-20-15-1 architecture from [Elsalamon & Elsayad, 2013]. There were also certain model instances, which show higher accuracy (ACC_{max} in the table of Figure 3). Models with 41-8-1 architecture have best average AUC of 0.915. Certain model instances achieved AUC=0.939. The variance and instability of results can be explained by insufficient saturation of data for training. The model can't be trained well to discriminate between classes, particularly to recognize the under-presented 'yes' class. Nevertheless, experiments show that given the data saturation, a 41-3-1 neural net can be trained to reach accuracy 93.143%, which significantly outperforms the 48-20-15-1 one proposed in [Elsalamon & Elsayad, 2013] and [Elsalamony, 2014].

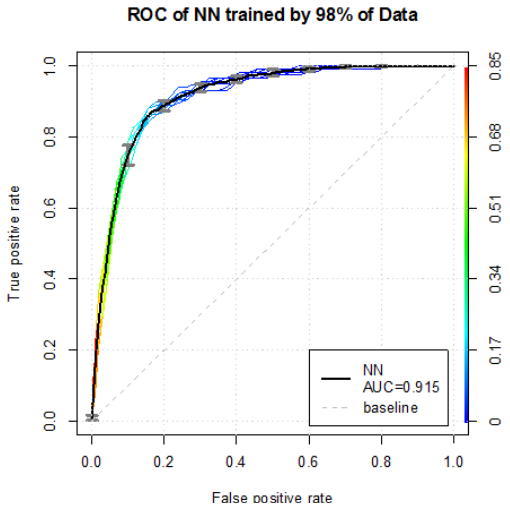

Fig. 4. ROC curves of 10 neural network models with 42-8-1 architecture. Colors sho Fig. 4. ROC curves of 10 neural network models with 42-8-1 architecture. Colors show values of the cutoff points applied. Black line represents average values of the 10 models. Standard deviation bars measure variance.

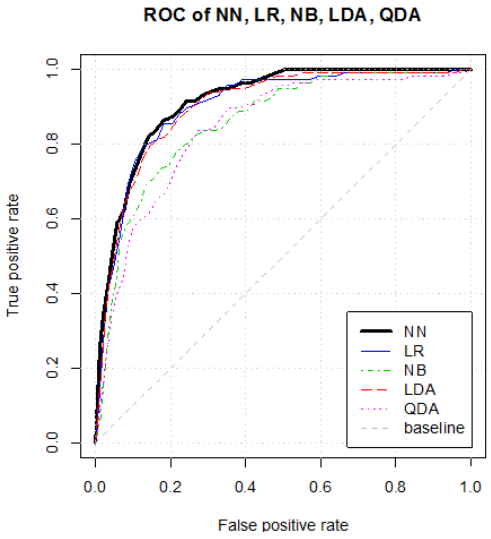

Fig. 5. ROC curve of 5 models: Neural Network, Logistic Regression, Naive Bayes, Linear Discriminant Analysis, and Quadratic Discriminant Analysis. Each model runs with its optimal hyper-parameter values and size of the training dataset

Figure 4 shows ten colored curves, each of which is a plot of a 41-8-1 NN model trained and validated by the 98% dataset. The colors represent different cutoff points with color bar shown on the right side of the box. The black curve is average of the 10 curves. The variance of TPR is depicted by the standard deviation bars.

In order to compare different modeling techniques, we also built models based on logistic regression, naive Bayes, linear discriminant analysis, and quadratic discriminant analysis. Figure 5 shows ROC curves of the models in one plot. Generally, if two ROC curves do not intersect then one model dominates over the other. When ROC curves cross each other, one model is better for some threshold values, and is worse for others. In that situation the AUC measure can lead to biased results and we are not able to select the best model. The figure shows that the curves of NN and LR intersect one another, but in most of the regions NN outperform LR being closer to the top-left corner. This is particularly visible in the most north-west regions, there maximal accuracy is achieved. NN entirely dominate over LDA, NB, and QDA, which performance can be ranked in that order.

Table 2 shows how data saturation affects performance of all 5 models. It can be seen that NN is best performer at nearly all levels of saturation with exception of poorly saturated data (10-20%), where LDA shows better characteristics, particularly measured by AUC.

Table 2: Performance of the five models with different levels of data saturation, ranging from 98% to 10% of the original dataset.

Model	NN		LR		NB		LDA		QDA	
% of dataset	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
98%	**90.489**	**0.915**	89.810	0.902	87.912	0.852	89.810	0.900	86.913	0.838
80%	**90.401**	**0.912**	89.510	0.896	88.212	0.853	89.710	0.900	87.213	0.835
60%	**90.342**	**0.910**	89.810	0.898	88.312	0.858	89.810	0.900	87.013	0.845
40%	**90.213**	**0.902**	89.710	0.895	86.813	0.847	89.910	0.901	86.813	0.831
20%	90.209	0.895	**90.210**	0.892	87.313	0.850	89.910	**0.898**	86.813	0.837
10%	**89.710**	0.893	89.710	0.889	87.712	0.844	89.610	**0.896**	86.313	0.826

Conclusion

This paper presents a case study of data mining modeling techniques for direct marketing. We address some issues which we find as gaps in previous studies, namely:

The most common partitioning procedure for training, validation, and test sets uses random sampling. Although, this is a fair way to select a sample, some 'lucky' draws train the model much better than others. Thus, the model instances show variance in behavior and characteristics, influenced by the randomness. In order to address this issue and further to [Moro et al., 2011], [Elsalamon & Elsayad, 2013], [Elsalamony, 2014], we used a methodology, which combines cross-validation (CV), multiple runs over random selection of the folds and initial weights, and multiple runs over random selection of partitions. Each model was tested 300 times involving 3-fold cross-validation, random partitioning and iterations. We applied double-testing with both validation and test sets.

We also explored how NN design affect the model performance in order to find the optimal size of the hidden layer. Given, that the task is a classic binary classification problem without clearly separable feature extraction stages, we found that the two-hidden layers architecture proposed in [Elsalamon & Elsayad, 2013], [Elsalamony, 2014] could be simplified to one hidden layer with structure 41-8-1. The simpler architectures are always preferable as they can be built and trained easily and run faster.

We also did comparatative analysis of neural nets, logistic regression, naive Bayes, linear and quadratic discriminant analysis taking into account their performance at different levels of data saturation. We found that NN is best performer in nearly all levels of saturation with exception of poorly saturated data (10-20%), where LDA shows better characteristics, measured by AUC. We also did comparatative ROC analysis of the models.

Bibliography

[Bache, & Lichman, 2013] Bache, K. & Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[Broyden , 1970] Broyden, C., The convergence of a class of double rank minimization algorithms: The new algorithm, J. Inst. Math. Appl., 6: 222–231, 1970.

[Chapman et al., 2000] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. CRISP-DM 1.0 - Step-by-step data mining guide, CRISP-DM Consortium, 2000

[Clark & Niblett, 1989] Clark, P. & Niblett, T. The CN2 induction algorithm. Machine Learning, 3, 261-283. 1989.

[Cortez, 2010] Cortez, P. "Data Mining with Neural Networks and Support Vector Machines using the R/rminer Tool". In Proc. of the 10th Industrial Conference on Data Mining (Berlin, Germany, Jul.). Springer, LNAI 6171, 572– 583, 2010.

[Elsalamon & Elsayad, 2013] Elsalamony, H., Elsayad, A., Bank Direct Marketing Based on Neural Network, International Journal of Engineering and Advanced Technology, 2(6):392-400, 2013.

[Elsalamony, 2014] Elsalamony, H. Bank Direct Marketing Analysis of Data Mining Techniques., International Journal of Computer Applications, 85 (7):12-22, 2014.

[Fawcett , 2005] Fawcett, T., An introduction to ROC analysis, Pattern Recognition Letters 27, No.8, 861–874, 2005

[Fisher, 1936] Fisher, R. "The Use of Multiple Measurements in Taxonomic Problems," Annals of Eugenics, 7, 179–188.

[Moro et al., 2011] Moro, S., Laureano, R., Cortez, P., Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011.

[Fletcher , 1970] Fletcher, R., A new approach to variable metric algorithms, Computer J., 13: 317–322, 1970.

[R Development Core Team, 2009] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org, 2009.

[Shin & Cho, 2006] Shin, H. J. and Cho, S., Response modeling with support vector machines, Expert Systems with Applications, 30(4): 746-760, 2006.

[Sing et al., 2005] Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T., ROCR: visualizing classifier performance in R., Bioinformatics 21(20):3940-3941, 2005.

[Sing'oei & Wang, 2013] Sing'oei, L., Wang, J., Data Mining Framework for Direct Marketing: A Case Study of Bank Marketing, International Journal of Computer Science Issues (IJCSI), 10(2):198-203, 2013.

[Yu & Cho, 2006] Yu, E. and Cho, S., Constructing response model using ensemble based on feature subset selection, Expert Systems with Applications, 30(2): 352-360, 2006.

Authors' Information

Anatoli Nachev – Business Information Systems, Cairnes Business School, National University of Ireland, Galway, Ireland; e-mail: anatoli.nachev@nuigalway.ie

Major Fields of Scientific Research: data mining, neural networks, support vector machines, adaptive resonance theory.