International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-4 Issue-4, April 2015

Using Support Vector Machines for Direct Marketing Models

A. Nachev, T. Teodosiev

Abstract — This paper presents a case study of data mining modeling for direct marketing, based on support vector machines. We address some gaps in previous studies, namely: dealing with randomness and 'lucky' set composition; role of variable selection, data saturation, and controlling the problem of under-fitting and over-fitting; and selection of kernel function and model hyper-parameters for optimal performance. In order to avoid overestimation of the model performance, we applied a double-testing procedure, which combines cross-validation, and multiple runs. To illustrate the points discussed, we built predictive models, which outperform those discussed in previous studies.

Index Terms — classification, data mining, direct marketing, support vector machines.

I. 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, and 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 [9], [10], [11], [12], and regarded as an efficient modeling technique. Decision trees have been explored in [10] and [11]. Support vector machines are also well performing models discussed in [9], [12], and [13]. Many other modeling techniques and approaches, both statistical and machine learning, have been studied and used in the domain.

Manuscript published on 30 April 2015.
***** Correspondence Author (s)
A. Nachev, BIS, Cairnes Business School, NUI Galway, Galway, Ireland.
T. Teodosiev, Department of Computer Science, Shumen University, Shumen, Bulgaria.

This paper focuses to support vector machines as a modeling technique and discuss factors, which affect their performance and capabilities to predict.

We extend the methodology used in [9], [10], and [11], addressing certain gaps, which influence model performance. The remainder of the paper is organized as follows: section II provides an overview of the data mining technique used; section III discusses the dataset used in the study, its features, and the preprocessing steps needed to prepare the data for experiments; section IV presents and discuses the experimental results; and section V gives conclusions.

II. SUPPORT VECTOR MACHINES

Support vector machines are common machine learning techniques. They belong to the family of generalized linear models, which achieve a classification or regression decision based on the value of the linear combination of input features. Using historical data along with supervised learning algorithms, SVM generate mathematical functions to map input variables to desired outputs for classification or regression prediction problems.

SVM, originally introduced by Vapnik [1], provide a new approach to the problem of pattern recognition with clear connections to the underlying statistical learning theory. They differ radically from comparable approaches such as neural networks because SVM training always finds a global minimum in contrast to the neural networks. SVM can be formalized as follows:

Training data is a set of points of the form

$$D = \{(x_i, c_i) \mid x_i \hat{I} \hat{A}^p, c_{i\hat{I}} \{-1,1\}\}_{i=1}^{n} \qquad (1)$$

where the c_i is either 1 or -1, indicating the class to which the point x_i belongs. Each data point x_i is a p-dimensional real vector. During training a linear SVM constructs a p-1-dimensional hyper-plane that separates the points into two classes (Fig. 1). Any hyper-plane can be represented by $w \times x - b = 0$, where w is a normal vector and $\cdot$ denotes dot product. Among all possible hyper-planes that might classify the data, SVM selects one with maximal distance (margin) to the nearest data points (support vectors).

When the classes are not linearly separable (there is no hyperplane that can split the two classes), a variant of SVM, called soft-margin SVM, chooses a hyperplane that splits the points as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The method introduces slack variables, X_i, which measure the degree of misclassification of the datum x_i. Soft-margin SVM penalizes misclassification errors and employs a parameter (the soft-margin cost constant C) to control the cost of misclassification.

Training a linear SVM classier solves the constrained optimization problem:

$$\min_{w,b,x_k} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} x_i \qquad (2)$$
$$s.t. \quad w \cdot x + b \geq 1 - x_i$$

In dual form the optimization problem can be represented by

$$\min_{a_i} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j x_i \cdot x_j - \sum_{i=1}^{n} a_i \qquad (3)$$
$$s.t. \quad 0 \leq a_i \leq C, \quad \sum_{i=1}^{n} a_i c_i = 0$$

The resulting decision function $f(x) = w \cdot x + b$ has a weight vector $w = \sum_{k=1}^{n} a_k y_k x_k$. Data points x_i for which $a_i > 0$ are called support vectors, since they uniquely define the maximum margin hyperplane. Maximizing the margin allows one to minimize bounds on generalization error.

If every dot product is replaced by a non-linear kernel function, it transforms the feature space into a higher-dimensional one, thus though the classifier is a hyperplane in the high-dimensional feature space (Fig. 2). The resulting classifier fits the maximum-margin hyperplane in the transformed feature space. The kernel function can be defined:

$$k(x_i, x_j) = \Phi(x_i)\Phi(x_j) \qquad (4)$$

where $\Phi(x)$ maps the vector x to some other Euclidean space. The dot product $x_i \cdot x_j$ in the formulae above is replaced by $k(x_i, x_j)$ so that the SVM optimization problem in its dual form can be redefined as: maximize (in a_i)

$$\tilde{L}(a) = \sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j y_i y_j k(x_i, x_j) \qquad (5)$$
$$s.t. \sum_i a_i y_i = 0; \quad a_i \geq 0; \quad 1 \leq i \leq N$$

A non-linear SVM is largely characterized by the choice of its kernel, and SVMs thus link the problems they are designed for with a large body of existing work on kernel-based methods. Some common kernels functions include:

- Linear kernel:
$$k(x, x') = x \cdot x' \qquad (6)$$
- Polynomial kernel:
$$k(x, x') = (scale\, x \cdot x' + offset)^{degree} \qquad (7)$$
- Gaussian RBF kernel:
$$k(x, x') = \exp(-s\|x - x'\|^2) \qquad (8)$$
- Hyperbolic tangent kernel:
$$k(x, x') = \tanh(scale\, x \cdot x' + offset) \qquad (9)$$
- Laplacian kernel:
$$k(x, x') = \exp(-s\|x - x'\|) \qquad (10)$$

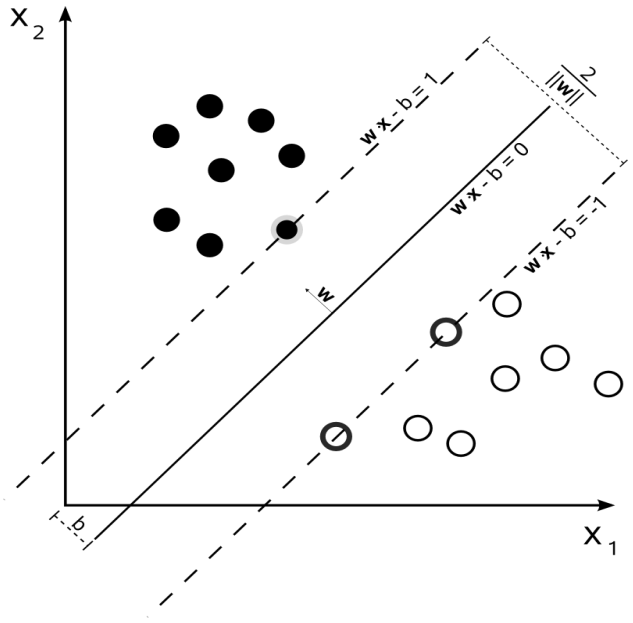

Figure 1. Maximum-margin hyperplane for a SVM trained with samples from two classes. Samples on the margin are support vectors.

The choice of kernel strongly depends on the task specifics and is usually made after empirical survey. The kernel parameters appear hyper-parameters for the model and their tuning is an impotrant for the classifier performance.

The SVM's major advantage lies with their ability to map variables onto an extremely high feature space. Because the size of the margin does not depend on the data dimension, SVM are robust with respect to data with high input dimension, however, it has been discovered they do not favor large datasets, due to the demands imposed on virtual memory, and the training complexity resultant from the use of such a scaled collection of data [2].

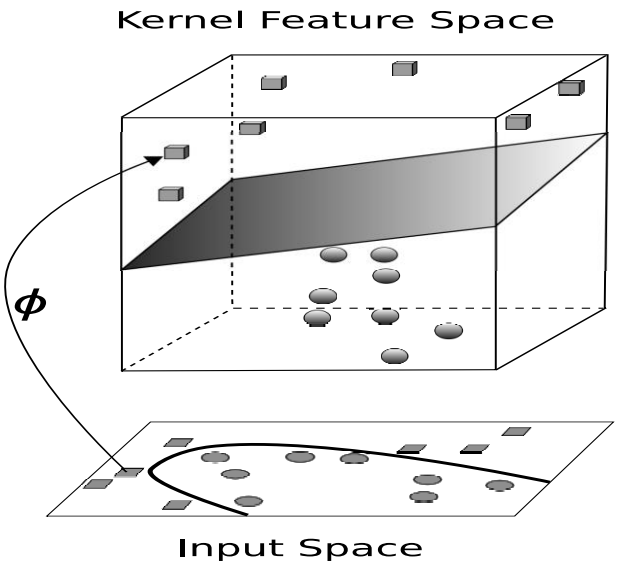

Figure 2. Kernel function: a linearly inseparable input space can be mapped to a linearly separable higher-dimmensional space.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Work from Fei, Li, and Yong [3] highlighted three "crucial problems" in the use of support vector machines. These are: attaining the optimal input subset, correct kernel function, and the optimal parameters of the selected kernel, all of which are prime considerations within this study.

III. DATA

A. Dataset

The direct marketing dataset used in this study was provided by Moro, Laureano, and Cortez [9], also available in [8]. It consists of 45,211 samples, each having 17 attributes, one of which is 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.)

The dataset is unbalanced, because the successful samples corresponding to the class 'yes' are 5,289, which is 11.7% of all samples. There are no missing values. Further details about data collection, understanding, and initial preprocessing steps can be found in [9].

With reference to the standard for data mining projects CRISP-DM [4], we did two data pre-processing transformations: mapping non-numeric data into binary dummies and normalization.

Non-numeric categorical variables were 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. However, in this example only three of the dummy variables are need - if values of three are known, the fourth is also known. For example, given that these four values are the only possible ones, we can know that if the education is neither unknown, primary, nor secondary, it must be tertiary. Thus we 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 [10] and [11]. There are two benefits of that: first, the model becomes simpler and faster; secondly, avoiding redundancy in data aliviates the SVM problem with demands imposed on virtual memory, and the training complexity with huge number of support vectors. The model building utility we used converts categorical variables to binary dummies without redundancy. The second data transformation we did is related to normalization/scaling. This procedure attempts to have all input variables x_a with consistent values, regardless of their original scale of and/or different measurement units used, e.g. *day* (1-31) vs. *duration* in seconds (0-4918). If the data are left as they are, the training process gets influenced and biased by some 'dominating' variables with large values. In order to address this, we did normalization (z-scoring) by:

$$x_{a,i}^{new} = \frac{x_{a,i} - m}{s} \quad i \in \{1,\ldots,N\} \tag{11}$$

where m is the mean and s is the standard deviation of x_a. After the transformation, each input variable has zero mean and unit standard deviation.

B. Variable Importance

Referring to the data preparation stage of the CRISP-DM project model for data mining [4], we explore how presence or absence of the input variables presented to the model for training and testing affects the classifier performance. Removing most irrelevant and redundant variables from the data may help to alleviate the effect of the curse of dimensionality, enhance the model generalization capabilities, speed up the learning process, and to improve the model interpretability. The variable selection also helps to acquire better understanding about data and how they are related with each other. This work uses Sensitivity Analysis (SA) for ranking the variable importance to the model by measuring the effects on the output when the inputs are varied through their range of values [5]. While initially proposed for neural nets, SA is currently used with virtually any supervised learning technique, such as SVM [6]. The SA varies an input variable x_a through its range with L levels, under a regular sequence from the minimum to the maximum value. Let $x_{a,j}$ denotes the j-th level of input x_a. Let $\hat{y}$ denote the value predicted by the model for one data sample (x) and let $\hat{y} = P(x)$ is the function of model responses. Kewley, Embrechts, and Breneman propose in [7] three sensitivity measures, namely range (S_r), gradient (S_g) and variance (S_v):

$$\begin{aligned} S_r &= \max(\hat{y}_{a_j} : j \in \{1,\ldots,L\}) - \min(\hat{y}_{a_j} : j \in \{1,\ldots,L\}) \\ S_g &= \sum_{j=2}^{L} \left|\hat{y}_{a_j} - \hat{y}_{a_{j-1}}\right| / (L-1) \\ S_v &= \sum_{j=2}^{L} (\hat{y}_{a_j} - \bar{y}_a)^2 / (L-1) \end{aligned} \tag{12}$$

where $\bar{y}_a$ denotes the mean of the responses. The gradient is the only measure that is dependent on the order of the sensitivity responses. For all measures, the higher the value, the more relevant is the input x_a. The relative importance r_a can be given by:

$$r_a = V_a / \sum_{i=1}^{M} V_i \tag{13}$$

where V_a is the sensitivity measure for x_a (e.g., range) [5].

IV. EXPERIMENTS AND DISCUSSION

In order to explore the SVM performance for task outlined and compare the model characteristics with those discussed in studies [9], [10], [11], we used the same dataset and did experiments consistently. Further to that, we extended the methodology addressing the following gaps:

- *Validation and testing.* Using validation and test sets in a double-testing procedure helps to avoid overestimation of the model performance.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

- *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 we solidify the conclusions made.
- *Choice of kernel function*. We explore the SVM performance using the five of the most commom kernel functions discussed above.
- *Optimization of the model hyper-parameters.* We tested the SVM performance with differen hyper-parameters, some of which are specific for the kernel function used.
- *Variable selection*. Further to identifying importance of variables and their contribution to the classification task on the basis of SA, we applied backward selection procedure to eliminate some input variables.
- *Data saturation*. We also explored the capacity of the SVM to act in early stages of data collection where lack of sufficient data may lead to underfitted models.

All experiments were conducted using R environment [15], [16], and [17].

In order to select input variables for elimination, we did SA using three sensitivity measures: range, gradient, and variance by 10 runs of the model per measure. Fig. 3-5 show the input variable importance, using a bar plot for each r_a in equation (13), sorted in descending order. The whiskers in the figures represent confidence intervals. Two of the measures, range and variance, find *loan* as the least significant input, while the gradient measure finds *default* the one. Anyway, both input variables show similar insignificance to the classification task. Applying backward variable selection procedure by eliminating first *loan*, we re-evaluated the input significances and further eliminated *contact* and *campaign* to obtain best results.

For the sake of consistency with the previous studies, we first used 98% of the original dataset, which was further split randomly into training and validation sets in ratio 2:1. The rest of 2% was used for final and independent test set. Using test set in addition to the validation set solidifies the performance estimation as the validation set specifics can influence the search for best hyper- parameters values. Thus, estimation based on validation set only can get biased.

In order to provide more realistic performance results and reduce the effect of lucky set composition, each version of the model was run 10 times with different randomly selected training and validation sets. For each run, a 3-fold cross-validation creates 3 model instances and averages their results. We iterated all those procedures 10 times per model, recording and averaging accuracy and AUC.

Another part of our experiments was to test how different levels of data saturation affect the SVM model performance. In a realistic situation, building a dataset can be an ongoing process, starting with a small dataset, which grow gradually over the time.

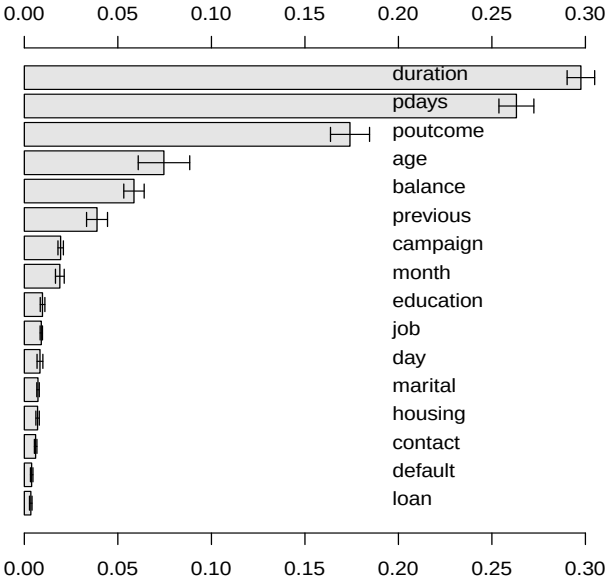

Figure 3. Input importances using 'range' sensitivity measure.

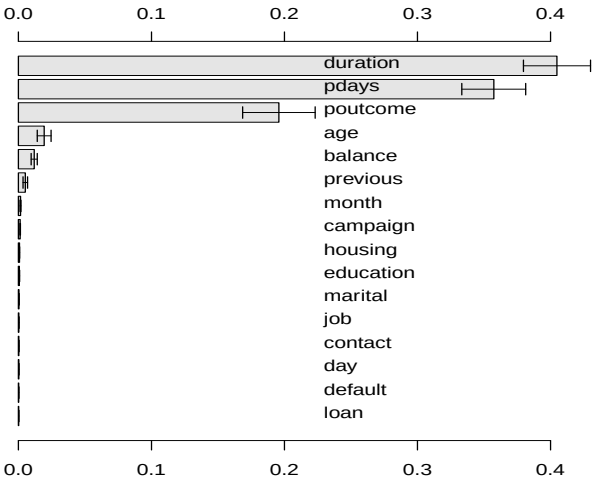

Figure 4. Input importances using 'variance' sensitivity measure.

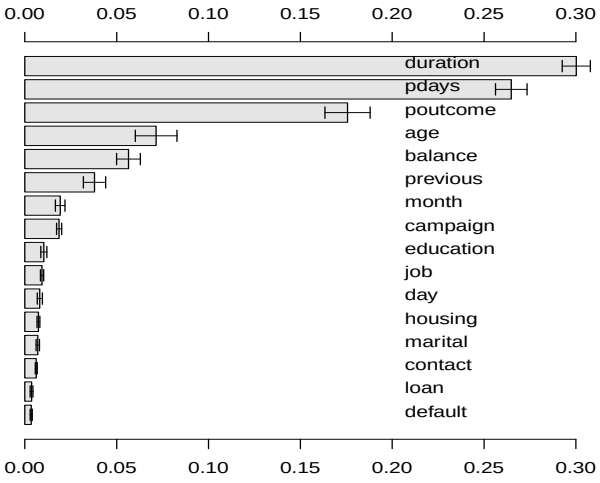

Figure 5. Input importances using 'gradient' sensitivity measure.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Performance of a classifier trained at different stages of the dataset lifetime is an important characteristic, as some modelling techniques may show better results than other in different data saturations. Table 1 summarizes the SVM performance in terms of accuracy and AUC with different levels of data saturation, ranging from 10% to 98% of the original dataset. Results show that the 20% saturation yields best average accuracy of 91.001% with some 'lucky sets' achieving 91.108%. The table also shows a dropping performance when data saturation gets higher / lower. This can be interpreted as having the size increasing / decreasing makes the model to over-fit / under-fit to the training set. The best model here outperforms the best models reported in previous studies [10], [11] with 90.09% max accuracy.

Table 1. SVM performance with fractions of the original dataset used for training.

Merit	*98%set*	*80%set*	*60%set*	*40%set*	*20%set*	*10%set*
ACC%	89.810	90.384	90.469	90.509	91.001	88.961
AUC	0.875	0.879	0.882	0.880	0.891	0.882

In data mining, classification performance is often measured using accuracy as the figure of merit. For a given operating point of a classifier, the accuracy is the total number of correctly classified instances divided by the total number of all available instances. Accuracy, however, varies dramatically depending on class prevalence. It can be a misleading estimator in cases where the most important class is typically underrepresented, such as the class of 'yes' of those who respond positively to the direct marketing. For these applications, sensitivity and specificity can be more relevant performance estimators. In order to address the accuracy deficiencies, we did Receiver Operating Characteristics (ROC) analysis [14]. In a ROC curve, the true positive rate (TPR), a.k.a. sensitivity, is plotted as a function of the false positive rate (FPR), a.k.a. 1-specificity, for different cut-off points. Each point on the ROC plot represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination between the two classes has a ROC plot that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC plot is to the upper left corner, the higher the overall accuracy of the test. The area under the ROC curve (AUC) is a common measure for the evaluation of discriminative power. AUC represents classifier performance over all possible threshold values, i.e. it is threshold independent.

We used the best performing 20% dataset for training and validation, internally split in into training and validation sets in ratio 2:1. The fit algorithm runs 10 times with different random selection of training and validation sets. For each of those set compositions, the 3-fold cross-validation creates 3 model instances and average results. Fig. 6 shows the results by 10 colored lines and a tick black curve, which is average of the 10 curves. Standard deviation bars, analogous to the whiskers, depict the variance of TPR.

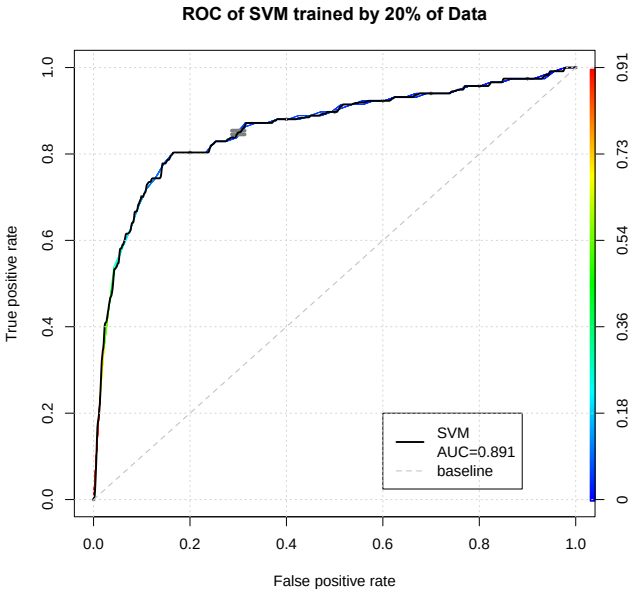

Figure 6. ROC curves of 10 SVM models. Black line represents average performance. Standard deviation bars measure variance.

Lift is another metric, often used to measure performance of marketing models. A good performance is when the response within the target is much better than average for the population as a whole. In a cumulative lift chart (gains chart), the y-axis shows the percentage of true positive responses (TPR). Formally,

$$TPR = sensitivity = TP/(TP+FN) \quad (14)$$

where TP and FN are true positive and false negative predictions, respectively. Fig. 7 shows the cumulative lift charts of the 10 SVM models, run under the ROC analysis. The colors and whiskers in the curves have the same purpose as above.

Another way to characterize performance of a classifier is to look at how precision and recall change as threshold changes. This can be visualized by precision-recall curve (Fig. 8). The better the classifier, the closer its curve is to the top-right corner of the graph. Formally,

$$precision = TP/(TP+FP) \quad (15)$$
$$recall = TP/(TP+FN) \quad (16)$$

In terms of a direct marketing task, precision is the percent of correctly identified 'yes' customers (who purchase the product) among all reported as 'yes'; recall is the percent of correctly identified 'yes' customers among those who are 'yes' in the test set. Recall and precision are inversely related: as recall increases, precision decreases and visa versa.

Another factor that affects the SVM performance is the choice of kernel function and selecting proper values for its parameters, which along with the misclassification cost C, hyper-parameters of the model. We explored empirically the SVM performance with the five kernels outlined in equations (6)-(10).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

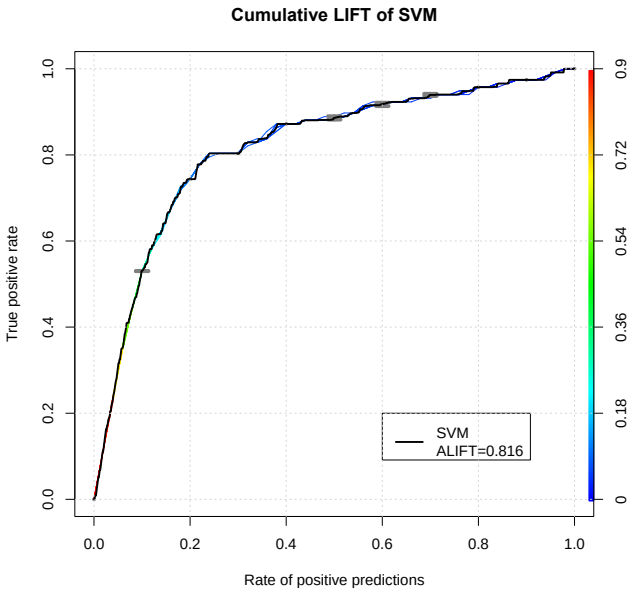

Figure 7. Cumulative LIFT curves of 10 SVM models. Black line represents the average. Standard deviation bars measure variance.

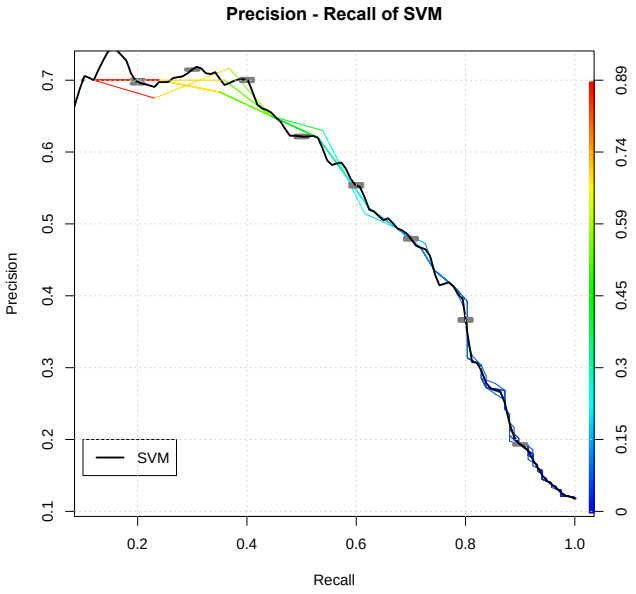

Figure 8. Precision-Recall curves of 10 SVM models. Black line represents the average. Standard deviation bars measure variance.

Table 2 summarizes outcomes. We found that Gaussian RBF is the best performing kernel in two sets of hyper-parameter values: C=3, sigma=0.089; C=3.5, sigma=0.091, both yeiding max ACC=91.108%. Fig. 9 illustrates grid search for optimal Gaussian RBF hyper-parameters in a range where the SVM provides the best discriminatory power. The highest two peaks correspond to the max accuracy obtained, while multiple ranges of both C and sigma obtain a very good accuracy above 90.7% outperforming the SVM models discussed in the previous studies. The training set here is the 20% of the original one and the three input variables *loan*, *contact*, and *campaign* were eliminated.

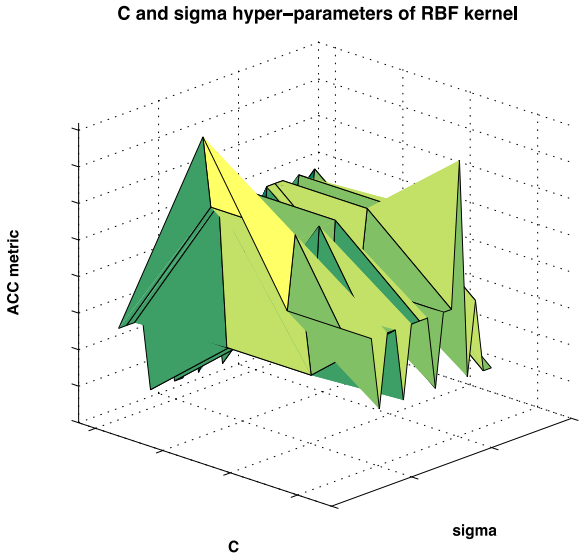

Figure 9. SVM performance with Gaussian RBF kernel and hyper-parameters C and sigma (σ).

Table 2. SVM optimal hyper-parameters using different kernel functions.

Hyper-parameter	*linear*	*RBF*	*poly*	*tanh*	*Laplacian*
C	1.4	3	3	3	2.75
sigma	n/a	0.089	n/a	n/a	0.083
degree	n/a	n/a	2	n/a	n/a
scale	n/a	n/a	1.5	3	n/a
offset	n/a	n/a	1	1.5	n/a
ACC$_{max}$%	89.210	**91.108**	88.911	88.411	90.109

Finally, we built Variable Effect Characteristic (VEC) curves [6] to explore the average impact of the four most important input variables x_a, which plot the x_{a_j} values (x-axis) versus the $\hat{y}_{a_j}$ responses (y-axis). Between two consecutive x_{a_j} values, the VEC plot uses a line (interpolation) for continuous values and a horizontal segment for categorical data. We run the model 10 times and plotted the average values vertically. The whiskers added represent the confidence intervals. Fig. 10-13 show how *duration*, *pdays*, *poutcome*, and *age* contribute the model performance.

From the *duration* VEC is evident that the last contact with shortest and longest duration contribute mostly to the positive outcome, whilst a moderate duration, typically about 2000 sec contributes to a negative outcome. Similarly, the *pdays* VEC shows that the sooner the customer is contacted after the last contact, the better. The gap between the contacts can be extended up to one year, but any over-delayed contact is useless and contributes to negative outcome.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

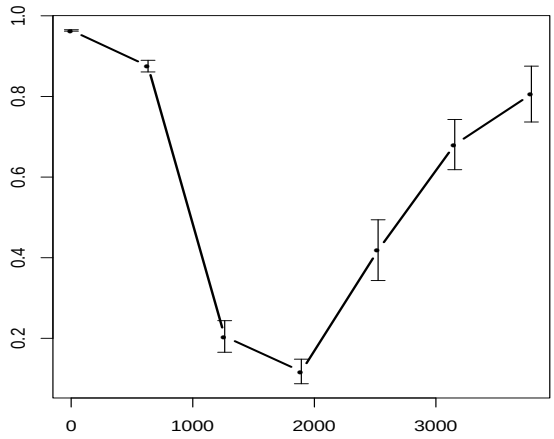

Figure 10. VEC curve for the 'duration' input.

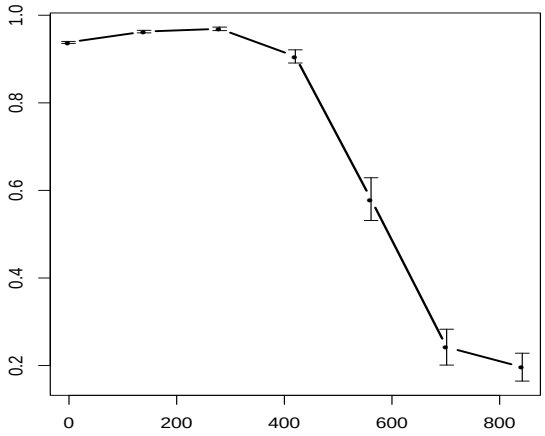

Figure 11. VEC curve for the 'pdays' input.

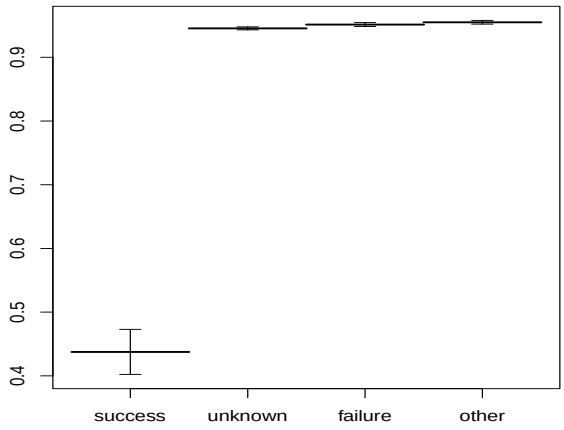

Figure 12. VEC curve for the 'poutcome' input.

In relation to the *poutcome* input, the VEC curve shows that customers who purchased the product or service are not likely to purchase it again and shouldn't be involved in the new direct marketing campaign, but there is a high chance to sell the product to any other customers.

Finally, the *age* VEC curve shows that the marketing campaign is better to target mid-age customers between 40-50; there is a negligible chance to sell the product to elderly people, particularly above 70.

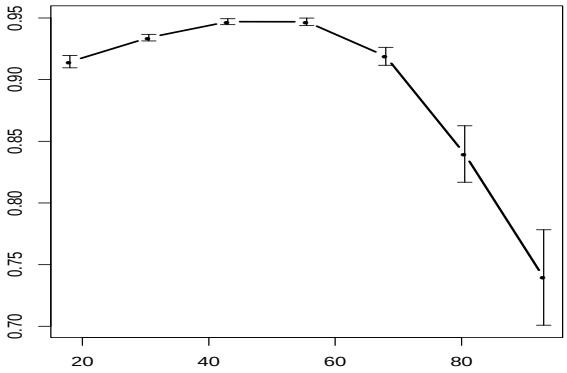

Figure 13. VEC curve for the 'age' input.

V. 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 [9], [10], [11], we used a methodology, which combines cross-validation (CV), multiple runs over random selection of the folds, and multiple runs over random selection of partitions. Each model was tested many times involving 3-fold cross-validation, random partitioning and iterations. We also applied double-testing procedure with both validation and test sets.

Another contribution is exploration of the SVM with different kernels and different values of hyper-parameters. The empirical results show that the best performing kernel is the Gaussian RBF with C=3, sigma=0.089; C=3.5, sigma=0.091, both yeiding max ACC=91.108%.

We also analysed how SVM performs with different levels of data saturation and found that the 20% dataset is best for training.

We also did analysis on how input variable importance affects the model performance and found that eliminating three inputs improve the SVM discriminatory power. Importance metrics were based on sensitivity analysis.

In conclusion, we believe that a rigorous model analysis, involving the issues discussed in the paper, lead to solid and better results.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[2] S. Horng, M. Su, Y. Chen, T. Kao, R. Chen, J. Lai, and C. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol.38, 2010, pp. 306-313.

[3] L. Fei, W. Li, and H. Yong, "Application of least squares support vector machines for discrimination of red wine using visible and near infrared spectroscopy," *Intelligent System and Knowledge Engineering*, vol. 1, 2008, pp. 1002-1006.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

[4] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "CRISP-DM 1.0 - Step-by-step data mining guide," *CRISP-DM Consortium*, 2000
[5] P. Cortez, M. Embrechts. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences* vol. 225, 2013, pp.1-17.
[6] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, 2009, pp. 547–553.
[7] R. Kewley, M. Embrechts, C. Breneman "Data strip mining for the virtual design of pharmaceuticals with neural networks," *IEEE Transactions on Neural Networks*, vol. 11 (3), 2000, pp. 668–679
[8] A. Asuncion and D. Newman, "UCI Machine Learning Repository, Univ. of California Irvine," [Online], Available: http://www.ics.uci.-edu/~mlearn/MLRepository.html.
[9] S. Moro, R. Laureano, P Cortez, "Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology," P. Novais (Ed.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, 2011, pp. 117-121.
[10] H. Elsalamony and A. Elsayad, "Bank Direct Marketing Based on Neural Network," *International Journal of Engineering and Advanced Technology*, vol. 2 no. 6, 2013, pp. 392-400.
[11] H. Elsalamony, "Bank Direct Marketing Analysis of Data Mining Techniques," *International Journal of Computer Applications*, vol. 85 no. 7, 2014, pp.12-22.
[12] E. Yu and S. Cho, "Constructing response model using ensemble based on feature subset selection", *Expert Systems with Applications*, vol. 30 no. 2, 2006, pp. 352-360.
[13] H. Shin and S. Cho, "Response modeling with support vector machines", *Expert Systems with Applications*, vol. 30 no. 4, 2006, pp. 746-760.
[14] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no.8, 2005, pp. 861–874.
[15] P. Cortez, "Data Mining with Neural Networks and Support Vector Machines using the R/rminer Tool." Proceedings of the 10th Industrial Conference on Data Mining, Springer, LNAI 6171, 2010, pp. 572–583.
[16] R Development Core Team. "R: A language and environment for statistical computing. R Foundation for Statistical Computing," [Online]. Available: http://www.R-project.org.
[17] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer, "ROCR: visualizing classifier performance in R," *Bioinformatics* vol. 21, no. 20, 2005, pp. 3940-3941.