International Journal of Pure and Applied Mathematics
Volume 119 No. 10 2018, 1149-1169
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

A comparison of machine learning techniques for customer churn prediction

Praveen Asthana
Canada

Abstract

We present a comparative study on the most popular machine learning methods applied to the challenging problem of customer churning prediction in the telecommunications industry. In the first phase of our experiments, all models were applied and evaluated using cross-validation on a popular, public domain dataset. In the second phase, the performance improvement offered by boosting was studied. In order to determine the most efficient parameter combinations we performed a series of Monte Carlo simulations for each method and for a wide range of parameters. Our results demonstrate clear superiority of the boosted versions of the models against the plain (non-boosted) versions. The best overall classifier was the SVM-POLY using AdaBoost with accuracy of almost 97% and F-measure over 84%.

Keywords:
Churn prediction, machine learning techniques, boosting algorithm

1. Introduction

Customer Relationship Management (CRM) is a comprehensive strategy for building, managing and strengthening loyal and long-lasting customer relationships. It is broadly acknowledged and extensively applied to different fields, e.g. telecommunications, banking and insurance, retail market, etc. One of its main objectives is customer retention. The importance of this objective is obvious, given the fact that the cost for customer acquisition is much greater than the cost of customer retention (in some cases it is 20 times more expensive [1]). Thus, tools to develop and apply customer retention models (churn models) are required and are essential Business Intelligence (BI) applications. In the dynamic market environment, churning could be the result of low-level customer satisfaction, aggressive competitive strategies, new products, regulations, etc. Churn models aim to identify early churn signals and recognize customers with an increased likelihood to leave voluntarily. Over the last decade there has been increasing interest for relevant studies in areas including telecommunication industry [2, 3, 4, 5, 6, 7, 8, 9, 10], banking [1, 11, 12, 13], insurance companies [14], and gaming [15], among others. Several, very popular in research community machine learning algorithms have been proposed in order to tackle the churning prediction problem. Such methods include Artificial Neural Networks [4, 5, 16, 17, 18], Decision Trees learning [4, 5, 7, 9, 16, 17], Regression Analysis [16], Logistic Regression [5, 9], Support Vector Machines [17], Naïve Bayes [4, 19], Sequential Pattern Mining and Market Basket Analysis [20], Linear Discriminant Analysis [13], and Rough Set Approach [21].

This work constitutes a comparison of five of the most widely used classification methods on the problem of customers' churning in the telecommunication sector. In particular, we compare the performance of multi-layer Artificial Neural Networks, Decision Trees, Support Vector Machines, Naïve Bayes classifiers, and Logistic Regression classifiers, compared to their boosting versions in an attempt to further improve their performance. The motivation behind our study is to evaluate the suitability of the state of the art machine learning methods on the problem of churning. This investigation is performed using Monte Carlo simulation at different settings of each classification method. We use the churn dataset originally from the UCI Machine Learning Repository (converted to MLC++ format[1]), which is now included in the package *C50* of the *R* language[2], in order to test the performance of classification methods and their boosting versions. Data are artificial based on claims similar to the real world. The dataset has been used in numerous publications [22, 23, 24, 25, 26, 27, 28].

The remainder of the paper is organized as follows. In Sec. 2, we give a brief presentation of the machine learning techniques that were evaluated. Evaluation criteria and the proposed boosting algorithm are presented in Sec. 3 and 4. The simulation setup and results are given in Sec. 5, and in Sec. 6 we draw our conclusions.

2. Machine Learning Techniques-Classification Methods

In the following, we briefly present five well established and popular techniques used for churn prediction, taking into consideration reliability, efficiency and popularity in the research community [4, 7, 9, 16, 17, 19, 29, 30].

2.1. Artificial Neural Network

Artificial Neural Networks (ANN) is a popular approach to address complex problems, such as the churn prediction problem. Neural networks can be hardware-based (neurons are represented by physical components) or software-based (computer models), and can use a variety of topologies and learning algorithms. One popular supervised model is the Multi-Layer Perceptron trained with variations of the Back-Propagation algorithm (BPN). BPN is a feed-forward model with supervised learning. In the case of the customer churn problem, Au e.a. [31] have shown that neural networks achieve better performance compared to Decision Trees. Also, experimental results showed that ANN outperformed Logistic Regression and C5.0 for churn prediction [30].

2.2. Support Vector Machines

Support Vector Machines (SVM), also known as Support Vector Networks, introduced by Boser, Guyon, and Vapnik [32], are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. SVM is a machine learning technique based on structural risk minimization. Kernel functions have been employed for improving performance [4]. Research on selecting the best kernels or combinations of kernels is still under way. In the churn prediction problem, SVM outperform DT and sometimes ANN, depending mainly on the type of data and data transformation that takes place among them [29, 17].

2.3. Decision Trees Learning

Decision Trees (DT) are tree-shaped structures representing sets of decisions capable to generate classification rules for a specific data set [33], or as Berry and Linoff noted *"a structure that can be used to divide up a large collection of records into successively smaller sets of records by applying a sequence of simple decision rules"* [34]. More descriptive names for such tree models are Classification Trees or Regression Trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. DT have no great performance on capturing complex and non-linear relationships between the attributes. Yet, in the customers churn problem, the accuracy of a DT can be high, depending on the form of the data [16].

2.4. Naïve Bayes

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naïve) independence assumptions. A more descriptive term for the underlying probability model would be independent feature model. In simple terms, a Naïve Bayes (NB) classifier assumes that the presence (or absence) of a particular feature of a class (i.e., customer churn) is unrelated to the presence (or absence) of any other feature. The NB classifier achieved good results on the churn prediction problem for the wireless telecommunications industry [19] and it can also achieve improved prediction rates compared to other widely used algorithms, such as DT-C4.5 [4].

2.5. Regression Analysis-Logistic Regression Analysis

Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. In terms of customer churning, regression analysis is not widely used, and that is because linear regression models are useful for the prediction of continuous values. On the other hand, Logistic Regression or Logit Regression analysis (LR) is a type of probabilistic statistical classification model. It is also used to produce a binary prediction of a categorical variable (e.g. customer churn) which depends on one or more predictor variables (e.g. customers' features). In the

		Predicted class	
		Churners	Non-Churners
Actual Class	Churners	TP	FN
	Non-Churners	FP	TN

Table 1: Confusion matrix for classifier evaluation

churn prediction problem, LR is usually used after proper data transformation is applied on initial data, with quite good performance [9] and sometimes performs as well as DT [7].

3. Evaluation Measures

In order to evaluate classifiers performance in churn prediction for different schemes with their appropriate parameters, we use the measures of *precision*, *recall*, *accuracy* and *F-measure*, calculated from the contents of the confusion matrix, shown in Tab. 1. True positive and false positive cases are denoted as TP and FP, respectively, while true negative and false negative cases are denoted as TN and FN.

Precision is the proportion of the predicted positive cases that were correct and is calculated from the equation

$$Precision = \frac{TP}{TP + FP}. \tag{1}$$

Recall is the proportion of positive cases that were correctly identified and is calculated from the equation

$$Recall = \frac{TP}{TP + FN}. \tag{2}$$

Accuracy is the proportion of the total number of predictions that were correct and is calculated from the equation

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{3}$$

Precision or recall alone cannot describe the efficiency of a classifier since good performance in one of those indices does not necessarily imply good performance on the other. For this reason, F-measure, a popular combination is commonly used as a single metric for evaluating classifier performance. F-measure is defined as the the harmonic mean of precision and recall

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{4}$$

A value closer to one implies that a *better combined precision and recall* is achieved by the classifier [35].

4. Boosting Algorithm

The main goal of boosting is to improve classification performance through the combination of decisions from many classification models, which are called weak classifiers (or weak learning algorithms). Boosting has been successfully applied to the prediction of customer churn in retail [36] and telecommunications companies [37]. In this paper, the effectiveness of boosting on a classifier will be measured by its ability to improve the respective F-measure. The weak classifiers are used as subroutines combined together in order to build an extremely accurate classifier in the train set. For each weak learner r, the boosting algorithm maintains a distribution of weights $w_r(i)$ on the training patterns i, so that each pattern can potentially have different contribution to the final training error of the learner. Initially all weights are set equally, but on each iteration, the weights of incorrectly classified elements are increased in order to force the weak classifier to focus on these hard cases. These continuously changing weights are called adaptive weights. Once the process has finished, the single classifiers obtained are combined into a final, theoretically highly accurate classifier in the train set. The final classifier therefore usually achieves a high degree of accuracy in the test set [11, 38, 39, 40].

There are a lot of forms of the boosting algorithms [19, 41], but the most popular is *AdaBoost*, where the weak classifiers are decision trees [42]. In this work, we use the *AdaBoost.M1* with DT and BPN as weak classifiers.

In more detail the AdaBoost.M1 is working as follows: For a training set

$$TS_n = [(x_1, y_1), \ldots, (x_n, y_n)] \tag{5}$$

with labels $y_i \in Y = [1, \ldots, l]$, a weight

$$w_r(i) = \frac{1}{n} \tag{6}$$

is initially assigned to every observation. These weights are recomputed, afterwards, according to weak classifier achievements. Iteratively, for $r = 1, \dots, k$, a weak classifier $C_r(x)$ is trained on TS_n in order to minimize the following error

$$E_r = \sum_{i=1}^{n} w_r(i)\, I(C_r(x_i) \neq y_i), \tag{7}$$

where I is the indicator function, equal to one when its argument is true, zero otherwise. After r iterations the weights are initially updated as follows

$$w_{r+1}(i) = w_r(i) \exp(a_r\, I(C_r(x_i) \neq y_i), \tag{8}$$

where

$$a_r = \frac{1}{2} \ln\left(\frac{1 - e_r}{e_r}\right) \quad \text{and} \quad e_r = \frac{E_r}{\sum_{i=1}^{n} w_i(i)}. \tag{9}$$

After the initial update the weights are re-normalized. The final *boosted* classifier is

$$C_{\text{final}}(x) = \underset{j \in Y}{\operatorname{argmax}} \sum_{r=1}^{k} a_r\, I(C_r(x_i) = j). \tag{10}$$

5. Cross Validation

5.1. Simulation Setup

Our main objective is to compare the most prominent classification methods on churn prediction. To that end, we implemented our simulation in two steps using the R language. In the first step, all tested classifiers are initially applied on the churning data. Their performance is evaluated using the F-measure criterion, discussed in Sec. 3. In the second step, a boosting algorithm is applied on the classifiers under inspection and their performance is measured again. For cross validation of our results, we generate 100 Monte Carlo realizations for different parameter scenarios in each classifier. The idea of Monte Carlo is the generation of a large number of synthetic data sets that are similar to experimental data. It is suggested to perform a *test* Monte Carlo with a small (e.g. 10-100) number of iterations in order to check the performance of the procedure. The tested classifiers were BPN, SVM, DT, NB and LR models. We employed boosting techniques in order to improve classification performance in three cases of ensemble models of BPN, SVM and DT. Ensembles of NB or LR models models were not formed since these models have no free parameters to tune. Therefore, no boosting has been applied in these cases.

Below, we give a brief presentation of the parameters of each classifier used in the simulation. The combination of all the values of parameters for each classifier and a size of 100 Monte Carlo realizations for each case, gives overall 5000 cases.

5.1.1. Artificial Neural Networks

The classic Back-Propagation algorithm is chosen to train a Multilayer Perceptron model with a single hidden layer (BPN). The number of hidden neurons, n, varies as $n = 5 \times i$, for $i = 1, 2, \ldots, 9$.

5.1.2. Support Vector Machines

The definition of simulation parameters for SVM varies depending on the kernel function that is chosen. In our simulations we chose the Gaussian Radial Basis kernel function (SMV-RBF) and the Polynomial kernel (SMV-POLY). For the RBF kernel,

$$K(x, y) = \exp\left\{\frac{-||x - y||^2}{2\,\sigma^2}\right\}, \tag{11}$$

σ varies as $\sigma = 0.001, 0.01, 0.1$, and the constant C, as $C = 10, 20, 30, 40, 50, 100$. For the Polynomial kernel,

$$K(x, y) = \left[x^T y + \theta\right]^p, \tag{12}$$

- θ takes the values $\theta = -2, -1, 0, 1, 2$, although $\theta = 1$ is preferable as it avoids problems with the Hessian becoming zero,
- the polynomial degree takes the values $p = 2, 3, 4, 5$,
- constant $C = 100$.

5.1.3. Decision Trees

We choose the default C5.0 algorithm for the DT classification (DT-C5.0) the follow up version of the well-known C4.5 [43]. C5.0 algorithm supports boosting, a method that improves the overall construction of the trees and gives more accuracy.

5.1.4. Naïve Bayes

In the NB model the conditional (posterior) probabilities of a categorical class variable given independent predictor variables are computed, using the Bayes rule.

5.1.5. Logistic Regression

In the LR model case the conditional (posterior) probabilities of a discrete-value variable are computed, given discrete or continuous variables.

5.2. Simulation Results

To evaluate the performance of tested classifiers, we use the churn dataset from the UCI Machine Learning Repository, which is now included in the package *C50* of the *R* language for statistical computing. In the corresponding version of the dataset there are 19 predictors, mostly numerical (num), plus the binary (yes/no) `churn` variable. In our study we do not consider the categorical `state` variable, since it's quite specific and we are interested in a more general approach. Thus, we consider 18 predictors, plus the `churn` variable, presented in Tab. 2.

The dataset contains a total of 5000 samples. In order to eliminate the bias, a 100-fold cross validation method is used. The training and testing data sets are randomly chosen with cross validation in a ratio of 2/3 and 1/3, respectively, for every Monte Carlo realization.

5.2.1. Classifiers performance without boosting

For the BPN case of the ANN classifier, our simulation results showed that the use of 20 neurons (or less) in the hidden layer, achieves better precision and quite good recall compared to other cases (Tab. 3). Especially, in the case of 15 hidden neurons, the average F-measure on 100 Monte Carlo realizations is 77.48% and it has a downward trend as the size of hidden layer increases. A similar behavior is also observed for the recall and accuracy metrics where the largest values, 71.68% and 94.06%, respectively, are achieved for the same number of hidden neurons.

These results are considered satisfactory, because the use of such size of hidden neurons minimizes the risk of overfitting.

For the RBF case of the SVM classifier, the highest values for all evaluation measures are found for $\sigma = 0.01$ and $C \geq 40$ (Tab. 4). Especially, in the case of $C = 40$, the F-measure becomes 73.16% on average, and the recall is 68.84%. The accuracy measure is 93.18% when $C = 30$. In the SMV-POLY

Variable Name	Type
`account_length` (number of months active user)	Num
`total_eve_charge` (total charge of evening calls)	Num
`area_code`	Num
`total_night_minutes` (total minutes of night calls)	Num
`international_plan`	Yes/No
`total_night_calls` (total number of night calls)	Num
`voice_mail_plan`	Yes/No
`total_night_charge` (total charge of night calls)	Num
`number_vmail_messages` (number of voice-mail messages)	Num
`total_intl_minutes` (total minutes of international calls)	Num
`total_day_minutes` (total minutes of day calls)	Num
`total_intl_calls` (total number of international calls)	Num
`total_day_calls` (total number of day calls)	Num
`total_intl_charge` (total charge of international calls)	Num
`total_day_charge` (total charge of day calls)	Num
`number_customer_service_calls` (number of calls to customer service)	Num
`total_eve_minutes` (total minutes of evening calls)	Num
`total_eve_calls` (total number of evening calls)	Num
`churn` (customer churn - target variable)	Yes/No

Table 2: Variables' names and types of the churn dataset. The variable `churn` is a Boolean variable indicating the target.

Neurons	Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
5	87.41	66.29	93.89	75.40
10	85.58	67.42	93.75	75.42
15	84.31	71.68	**94.06**	**77.48**
20	82.60	71.32	93.78	76.55
25	80.26	70.33	93.25	74.97
30	78.99	69.60	93.02	74.00
35	77.05	68.56	92.68	72.56
40	78.42	68.94	92.92	73.37
45	77.53	68.86	92.75	72.94

Table 3: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of BPN. (Highest values in bold.)

σ	C	Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
0.001	10	2.10	0.00	85.94	-
0.001	20	2.20	0.06	85.94	-
0.001	30	86.36	33.66	89.90	48.44
0.001	40	81.15	51.04	91.43	62.67
0.001	50	89.73	20.38	88.32	33.21
0.001	100	86.01	32.71	89.66	47.39
0.01	10	91.03	24.74	89.04	38.91
0.01	20	85.14	51.59	91.91	64.25
0.01	30	84.79	62.81	**93.18**	72.16
0.01	40	78.05	68.84	92.88	**73.16**
0.01	50	85.10	59.75	92.78	70.21
0.01	100	85.01	62.96	93.15	72.34
0.1	10	88.97	50.32	92.12	64.28
0.1	20	78.11	60.90	92.09	68.44
0.1	30	67.56	58.00	90.17	62.42
0.1	40	67.14	57.93	90.09	62.19
0.1	50	69.41	58.10	90.39	63.25
0.1	100	67.93	57.83	90.11	62.47

Table 4: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of SVM-RBF. (Highest values in bold.)

case, the highest values for most evaluation measures are found for $\theta = 1$ (recall, accuracy and F-measure) and for $p \geq 4$ (Tab. 5). Especially, for $\theta = 1$ and $p = 4$, the accuracy is 93.04% and the F-measure is 73.11% with the corresponding value of recall at 67.17%. For $p = 5$, recall achieves 68.46% with the corresponding value of F-measure at 72.41%, a value that is quite close to the maximum F-measure of SMV-POLY case.

For the DT case, the C5.0 algorithm achieves 77.04% on average for the F-measure and the corresponding recall and accuracy measures are 86.74% and 94.15% respectively, the largest evaluation measures compared to other classification methods (Tab. 6).

For the NB classifier, simulation showed that it is not as effective as the other classification methods (F-measure is 53.31% on average). For the LR classifier the results are quite similar with NB, where despite the high accuracy, 87.94%, the average value of F-measure is only 14.46% (for 100 Monte Carlo iterations), due to low precision and recall (Tab. 6).

θ	p	Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
-2	2	6.13	6.19	73.55	6.16
-1	2	6.23	6.29	73.57	6.26
0	2	72.89	41.45	89.57	52.84
1	2	78.37	54.70	91.49	64.43
2	2	78.38	54.72	91.50	64.45
-2	3	22.52	22.04	78.33	22.28
-1	3	21.93	21.37	78.21	21.64
0	3	88.05	47.54	91.71	61.74
1	3	83.47	62.71	93.00	71.62
2	3	83.20	62.86	92.98	71.62
-2	4	6.34	6.40	73.62	6.37
-1	4	6.72	6.75	73.75	6.74
0	4	85.68	16.40	87.84	27.53
1	4	80.21	67.17	**93.04**	**73.11**
2	4	77.60	68.39	92.77	72.71
-2	5	22.07	21.64	78.21	21.85
-1	5	20.58	20.22	77.78	20.40
0	5	99.42	2.83	86.33	5.51
1	5	76.85	68.46	92.65	72.41
2	5	69.54	67.60	91.26	68.55

Table 5: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of SVM-POLY. (Highest values in bold.)

Classifier	Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
DT-C5.0	69.29	86.74	94.15	77.04
NB	52.54	54.10	86.94	53.31
LR	8.05	70.55	87.94	14.46

Table 6: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of DT, NB, and LR.

To summarize, simulation results showed that BPN, with the number of hidden neurons of hidden layer to be less than 20, and DT-C5.0 classifiers are the most effective for the churn prediction problem on this specific dataset, with the SVM classifier to be very close. The performance of NB and LR classifiers fall short compared to BPN, SVM and DT.

5.2.2. Classifiers performance with boosting

Next, we apply the AdaBoost.M1 algorithm to explore the impact of boosting to the performance of the classifiers. Adaboost.M1 algorithm can only be applied to BNP, SVM and DT classifiers. Boosting requires the free model parameters to tune something that cannot be done in the case of NB or LR classifiers (the lack of parameters implies that the application of the classifier to the dataset several times will always give the same result).

For BPN with five internal weak learners, $k = 5$, accuracy is 95% and F-measure is nearly 81%. Similar results were obtained for ten internal weak learners, $k = 10$; accuracy is 94% and F-measure achieves 80.2% (Tab. 7) .

In order to apply boosting for the case of the SVM classifier we use different parameter sets of equal length to the size of the internal weak learners. For the SVM-RBF classifier we used two sets for the parameter C, $C = 10, 20, 30, 40, 50$ and $C = 60, 70, 80, 90, 100$. For the first set the F-measure is 76.18%, similar to the result without the use of boosting. For the second set, the improvement of performance is significant since accuracy is 96.05% and F-measure achieves 84.22%. For SVM-POLY classifier, tuning of the parameter θ does not result to high performance, since the achieved F-measure is only 40.84%. On the other hand, when the polynomial degree parameter p is tuned, then the boosting classifiers has significant improvement on its performance, since accuracy is 96.85% and the F-measure is 84.57%. The results are presented in Tab. 8.

In the DT case, the use of AdaBoost.M1 results in accuracy of 95.09% and the F-measure obtained is 83.87% (Tab. 9).

5.3. Performance comparison - Discussion

A performance comparison of the BPN, SVM and DT classifiers for the version with and without boosting is presented in Tab. 10. It is apparent that the use of boosting improves significantly the classifiers' performance, especially the F-measure metric. The improvement of the accuracy metric is 1% in the case of the BPN, 3% for SVM-RBF, 4% for SVM-POLY, and almost 1% for DT. For the F-measure the improvement is substantially higher,

Neurons	Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
		BPN AdaBoost.M1, k = 5		
5	87.41	66.97	93.88	75.42
10	85.20	65.68	93.48	73.75
15	84.00	78.17	**95.05**	**80.97**
		BPN AdaBoost.M1, k = 10		
5	87.76	67.10	93.94	75.67
10	86.81	74.49	94.24	80.18
15	84.13	70.57	93.93	76.53

Table 7: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of BPN with boosting. (Highest values in bold.)

	Parameters fitted	Precision	Recall	Accuracy	F-measure
	on weak classifier	(%)	(%)	(%)	(%)
SVM-RBF	C=10,20,30,40,50	93.94	64.16	94.37	76.18
AdaBoostM.1	C=60,70,80,90,100	96.19	74.99	**96.05**	**84.22**
SVM-POLY	θ=-2,-1,0,1,2	99.88	25.74	89.58	40.84
AdaBoostM.1	p=1,2,3,4,5	91.72	78.45	**96.85**	**84.57**

Table 8: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of SVM-RBF and SVM-POLY with boosting. (Highest values in bold.)

4.5% in the case of the BPN, more than 15% in the case of the SVM-RBF and SVM-POLY, and almost 9% in the case of DT. This improvement rate may be substantial in the case of a telecom provider where a typical database includes, on average, millions of customers. Our experiments suggest that a) the use of boosting can significantly improve the classification performance, and it is therefore recommended and b) SVM compare to the other methods is the most powerful tool for tackling the problem of customers churn prediction.

Precision (%)	Recall (%)	Accuracy (%)	F-measure (%)
	DT-C5.0 AdaBoost.M1		
77.60	90.07	95.09	83.87

Table 9: Precision, recall, accuracy and F-measure (estimated averages) for 100 Monte Carlo realizations of DT-C5.0 with boosting

Classifier	Accuracy (%)		F-measure (%)	
	Without Boosting	Boosting	Without Boosting	Boosting
BPN	94.06	95.05	77.48	80.97
SVM-RBF	93.18	96.05	73.16	84.22
SVM-POLY	93.04	96.85	73.11	84.57
DT-C5.0	94.15	95.09	77.04	83.87

Table 10: Performance comparison for BPN, SVM and DT classifiers with and without boosting.

6. Conclusion and Summary

Monte Carlo simulations were performed using five of the most popular, state-of-the-art classification methods for the churn prediction problem of telecom customers based on a publicly available dataset. Initially all methods were tested without the use of boosting under different settings. The two top performing methods in terms of corresponding testing error were the two-layer Back-Propagation Network with 15 hidden units and the Decision Tree classifier; both methods achieved accuracy 94% and F-measure 77%, approximately. The Support Vector Machines classifiers (RBF and POLY kernels) obtained accuracy of about 93% and an approximate F-measure of 73%. Naïve Bayes and Logistic Regression methods fail short with approximate accuracy 86% and an F-measure of about 53% and 14%, respectively.

Subsequently, we investigated the impact of the application of boosting to the corresponding classifiers using the AdaBoost.M1 algorithm. Naïve Bayes and Logistic Regression classifiers cannot be boosted due to the lack of free parameters to be tuned. Comparative results showed performance improvement for all three remaining classifiers due to boosting. Accuracy has been improved between 1% and 4%, while F-measure between 4.5% and 15%. Overall, the best classifier was the boosted SVM (SVM-POLY with AdaBoost) with accuracy of almost 97% and F-measure over 84%.

This work has shed some light on the performance of popular machine learning techniques for the churning prediction problem and supported the advantage of the application of boosting techniques. In future work we plan to explore additional simulation schemes for the parameters of the weak learners for the AdaBoost.M1 algorithm, and to explore the performance of additional boosting algorithms beyond AdaBoost. Also, to use a larger and more detailed data set from the telecom industry in order to maximize the statistical significance of our results.

References

[1] The Chartered Institute of Marketing, Cost of customer acquisition versus customer retention (2010).

[2] S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, A. Rehman, Telecommunication subscribers' churn prediction model using machine learning, in: Digital Information Management (ICDIM), 2013 Eighth International Conference on, IEEE, 2013, pp. 131–136.

[3] K. Kim, C.-H. Jun, J. Lee, Improved churn prediction in telecommunication industry by analyzing a large network, Expert Systems with Applications.

[4] C. Kirui, L. Hong, W. Cheruiyot, H. Kirui, Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining, International Journal of Computer Science Issues (IJCSI) 10 (2).

[5] G. Kraljević, S. Gotovac, Modeling data mining applications for prediction of prepaid churn in telecommunication services, AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije 51 (3) (2010) 275–283.

[6] R. J. Jadhav, U. T. Pawar, Churn prediction in telecommunication using data mining technology, IJACSA Editorial.

[7] D. Radosavljevik, P. van der Putten, K. K. Larsen, The impact of experimental setup in prepaid churn prediction for mobile telecommunications: What to predict, for whom and does the customer experience matter?, Trans. MLDM 3 (2) (2010) 80–99.

[8] Y. Richter, E. Yom-Tov, N. Slonim, Predicting customer churn in mobile networks through analysis of social groups., in: SDM, Vol. 2010, SIAM, 2010, pp. 732–741.

[9] Ş. Gürsoy, U. Tuğba, Customer churn analysis in telecommunication sector, Journal of the School of Business Administration, Istanbul University 39 (1) (2010) 35–49.

[10] K. Tsiptsis, A. Chorianopoulos, Data mining techniques in CRM: inside customer segmentation, John Wiley & Sons, 2011.

[11] F. Eichinger, D. D. Nauck, F. Klawonn, Sequence mining for customer behaviour predictions in telecommunications, in: Proceedings of the Workshop on Practical Data Mining at ECML/PKDD, 2006, pp. 3–10.

[12] U. D. Prasad, S. Madhavi, Prediction of churn behaviour of bank customers using data mining tools, Indian Journal of Marketing 42 (9) (2011) 25–30.

[13] Y. Xie, X. Li, Churn prediction with linear discriminant boosting algorithm, in: Machine Learning and Cybernetics, 2008 International Conference on, Vol. 1, IEEE, 2008, pp. 228–233.

[14] C.-C. Günther, I. F. Tvete, K. Aas, G. I. Sandnes, Ø. Borgan, Modelling and predicting customer churn from an insurance company, Scandinavian Actuarial Journal 2014 (1) (2014) 58–71.

[15] J. Kawale, A. Pal, J. Srivastava, Churn prediction in MMORPGs: A social influence based approach, in: Computational Science and Engineering, 2009. CSE'09. International Conference on, Vol. 4, IEEE, 2009, pp. 423–428.

[16] J. Hadden, A. Tiwari, R. Roy, D. Ruta, Churn prediction: Does technology matter, International Journal of Intelligent Technology 1 (2) (2006) 104–110.

[17] E. Shaaban, Y. Helmy, A. Khedr, M. Nasr, A proposed churn prediction model, Journal of Engineering Research and Applications 2 (4) (2012) 693–697.

[18] C.-P. Wei, I. Chiu, et al., Turning telecommunications call details to churn prediction: a data mining approach, Expert systems with applications 23 (2) (2002) 103–112.

[19] S. V. Nath, R. S. Behara, Customer churn analysis in the wireless industry: A data mining approach, in: Proceedings-Annual Meeting of the Decision Sciences Institute, 2003, pp. 505–510.

[20] M. Farquad, V. Ravi, S. B. Raju, Churn prediction using comprehensible support vector machine: An analytical crm application, Applied Soft Computing 19 (2014) 31–40.

[21] A. Amin, S. Shehzad, C. Khan, I. Ali, S. Anwar, Churn prediction in telecommunication industry using rough set approach, in: New Trends in Computational Collective Intelligence, Springer, 2015, pp. 83–95.

[22] M. Kuhn, K. Johnson, Applied predictive modeling, Springer, 2013.

[23] G. Klepac, Developing Churn Models Using Data Mining Techniques and Social Network Analysis, IGI Global, 2014.

[24] D. T. Larose, Discovering knowledge in data: an introduction to data mining, John Wiley & Sons, 2014.

[25] A. Sharma, D. P. K. Panigrahi, A neural network based approach for predicting customer churn in cellular network services, International Journal of Computer Applications 27 (11) (2011) 26–31.

[26] E. Antipov, E. Pokryshevskaya, Applying chaid for logistic regression diagnostics and classification accuracy improvement, Journal of Targeting, Measurement and Analysis for Marketing 18 (2) (2010) 109–117.

[27] N. Kerdprasop, P. Kongchai, K. Kerdprasop, Constraint mining in business intelligence: A case study of customer churn prediction, International Journal of Multimedia and Ubiquitous Engineering 8 (3).

[28] A. Herschtal, B. Raskutti, Optimising area under the roc curve using gradient descent, in: Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 49.

[29] Y. Hur, S. Lim, Customer churning prediction using support vector machines in online auto insurance service, in: Advances in Neural Networks–ISNN 2005, Springer, 2005, pp. 928–933.

[30] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, H. Kaushansky, Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry, Neural Networks, IEEE Transactions on 11 (3) (2000) 690–696.

[31] W.-H. Au, K. C. Chan, X. Yao, A novel evolutionary data mining algorithm with applications to churn prediction, Evolutionary Computation, IEEE Transactions on 7 (6) (2003) 532–545.

[32] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 144–152.

[33] S. J. Lee, K. Siau, A review of data mining techniques, Industrial Management & Data Systems 101 (1) (2001) 41–46.

[34] G. S. Linoff, M. J. Berry, Data mining techniques: for marketing, sales, and customer relationship management, John Wiley & Sons, 2011.

[35] T. Fawcett, An introduction to roc analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[36] M. Clemente, V. Giner-Bosch, S. San Matías, Assessing classification methods for churn prediction by composite indicators, Manuscript, Dept. of Applied Statistics, OR & Quality, Universitat Politcnica de Valncia, Camino de Vera s/n 46022.

[37] A. Lemmens, C. Croux, Bagging and boosting classification trees to predict churn, Journal of Marketing Research 43 (2) (2006) 276–286.

[38] J. Friedman, T. Hastie, R. Tibshirani, et al., Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The annals of statistics 28 (2) (2000) 337–407.

[39] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, Machine learning 36 (1-2) (1999) 105–139.

[40] R. E. Schapire, A brief introduction to boosting, in: Ijcai, Vol. 99, 1999, pp. 1401–1406.

[41] R. E. Schapire, Y. Freund, Boosting: Foundations and algorithms, MIT Press, 2012.

[42] Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: Computational learning theory, Springer, 1995, pp. 23–37.

[43] J. R. Quinlan, C4. 5: programs for machine learning, Vol. 1, Morgan kaufmann, 1993.