Proceedings of the 8th Polish Symposium of Physics in Economy and Social Sciences FENS, Rzeszów, November 4–6, 2015

Entropy Based Trees to Support Decision Making for Customer Churn Management

K. Gajowniczek*, A. Orłowski and T. Ząbkowski
Department of Informatics, Faculty of Applied Informatics and Mathematics, WULS-SGGW,
Nowoursynowska 159, 02-776 Warsaw, Poland

In this work we analyze empirically customer churn problem from a physical point of view to provide objective, data driven and significant answers to support decision making process in business application. In particular, we explore different entropy measures applied to decision trees and assess their performance from the business perspective using set of model quality measures often used in business practice. Additionally, the decision trees are compared with logistic regression and two machine learning methods — neural networks and support vector machines.

DOI: 10.12693/APhysPolA.129.971
PACS/topics: 05.45.Tp, 05.40.Ca, 07.05.Kf, 07.05.Mh

1. Introduction

Recent years resulted in the increasing automation and informatization of industry and enterprises that have been accumulating vast amounts of detailed and high-frequency data. These data provide opportunities from other fields, such as physics, mathematics, and information sciences, to gain insight into the business from the other perspective. Using novel empirical approaches for searching regularities and patterns akin to those in the physics, we believe to produce intriguing results.

Telecommunication industry is good example of the sector that easily accommodates all kinds of innovations contributing to business development. The companies have a unique advantage in marketplace by controlling the communications infrastructure, which generates more data than any other industry on customers and their usage behaviors. To be able to leverage, customer information telecom companies had to transform a traditional business model into one that meets today's demand for real-time business and consumer insight. To be feasible, it had to cope with the volume, variety, and complexity of data.

The churn problem, understood as substantial loss of valuable customers to competitors, is particularly severe since the telecoms are operating on fiercely competitive market. The customers demand new services and new technologies at lower prices, while telecommunication providers focus on acquisitions as their business goals, in many cases, not taking into account that retention strategy can play an important role for cellular providers. That is why companies suffer from a substantial loss of valuable customers, which is intensified due to market dynamics and its liberalization. Customers can freely choose among cellular service providers and actively migrate from one service provider to another. Apart from the above quoted scheme, referred as voluntary churn, there is also involuntary churn. Involuntary churn takes place when the company suspends the customer's service and this is usually due to non-payment or service abuse.

The motivation on churn is based on the fact that it costs more to recruit new customers than to retain existing ones, especially those high profitable customers. From this perspective we aim to deliver a data driven analysis of customer churn problem using different entropy measures applied to decision trees; additionally some other machine learning methods were used for comparison. In particular, we used real data from telecommunication industry to predict the churn event. While having it predicted, some retention actions can be usually addressed in advance to prevent the customer loss. For instance, when communicating the retention offer to the customer, certain services are proposed in order to make them stay. The following message may be the illustrative example for prepaid customers: make at least 50 PLN value recharge during next 10 days and you will receive additional 10 PLN for the calls.

This paper is a continuation of a study published in [1]. With respect to our previous work we significantly extended it by in depth analysis of classification trees properties and their complex comparison with other competitive machine learning methods. We argue that precise selection of entropy measure and its parameters can result in robust models with high ability to predict customer behavior. We believe that our research fits into attempt to generate value added to business and gain operational efficiency, that is targeting only a small group of clients with the highest churn probability.

2. Literature on techniques for churn classification

Customer churn prediction is a binary classification problem but due to the high data dimensionality and usually small number of minority class in the telecom

*corresponding author; e-mail: krzysztof_gajowniczek@sggw.pl

datasets, it makes a big hurdle for conventional classifiers to show desired performance. The scale of the problem justifies the need for its accurate identification and proposing some retention activities in advance. Researchers emphasize that an important role in the whole process strongly depends on the technique used, data type, and its quality.

In literature, most of the churn studies use data mining or statistical methods, in order to predict the probabilities of churn. The common approach is to use a set of techniques e.g. decision trees, logistic regression or neural networks and to compare their predictive power in order to determine the best technique for churn classification [2, 3]. The other two studies were focused on support vector machines performance for churn detection: Kim et al. [4] investigated how effectively support vector machines can detect churn in comparison to back-propagation neural networks for predicting on a data set from a credit card company; Zhao et al. [5] performed similar comparison analysis between one-class support vector machines, neural networks, decision trees and naive Bayes.

Zabkowski and Szczesny [6] demonstrated neural network and decision trees for customer insolvency (involuntary churn) in cellular telecommunications and the results proved that neural network models are more stable than decision trees. Nie et al. [7] used logistic regression and decision tree model and [8] focused on binomial logistic regression model for churn prediction and identified customer dissatisfaction, service usage, switching cost and demographic variable affects customer churn. Others [9] evaluated own churn prediction technique based on multi classifier class-combined approach that predicts churning from subscriber contractual information and call pattern changes.

In general, there are many practical application that use supervised learning techniques, such as: logit and probit [10–15] which are extension of classical regression methods, adapted specifically for the classification. In turn, [8, 9, 16, 17] have used decision trees, which are graphical decision support method used in decision theory. Others, including [3, 16, 18–21] have successfully used artificial neural networks.

From this perspective a comprehensive study of different approaches seems to be a reasonable way to analyze the underlying problem profoundly and to draw either general and specific conclusions on the topic.

3. Classification trees with entropy measures

Classification trees are powerful and very popular tools for multivariate variable analysis with their beginning around 60'ties of the 20th century. A very fast development of algorithms used in classification trees took place in eighties and nineties [22, 23]. Nowadays, classification trees are widely used and still being developed. The attractiveness of this technique is due to the fact that they create rules that can be easily interpretable. The tree attempts to find a strong relationship between input and target values in a group of observations using some discrimination measures. At each step of the tree construction, the discrimination power of each attribute with regard to the class is measured.

In this paper we are particularly interested in the application of entropy measures such as Shannon's [24], Tsallis's [25] and Rènyi's [26].

We assume that observations may belong to two given classes and for the classification we use a modified algorithm similar to C4.5 [23] to construct a binary tree in R environment [27].

As a general measure of diversity of objects, a Shannon entropy is often used which is defined as:

$$H_s = -\sum_{i=1}^{n} p_i \log p_i, \tag{1}$$

where p_i is the probability of occurrence of an event x_i being an element of the event X that can take values $x_i, ..., x_n$. The value of the entropy depends on two parameters: (1) disorder (uncertainty) and is maximal when the probability for every is equal; (2) the value of n. The Shannon entropy assumes a tradeoff between contributions from the main mass of the distribution and the tail. To control both parameters two generalizations were proposed by Rènyi and Tsallis.

The Rènyi entropy is defined as:

$$H_s = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^{\alpha} \right), \tag{2}$$

where parameter α is used to adjust the measure depending on the shape of probability distributions.

The Tsallis entropy is defined as:

$$H_s = \frac{1}{\alpha - 1} \left(1 - \sum_{i=1}^{n} p_i^{\alpha} \right). \tag{3}$$

With the Shannon entropy, events with high or low probability have equal weights in the entropy computation. However, using the Tsallis entropy, for $\alpha > 1$, events with high probability contribute more than low probabilities for the entropy value. Therefore, the higher is the value of α, the higher is the contribution of high probability events in the final result. Furthermore, increasing α parameter ($\alpha \to \infty$) makes the Rènyi entropy determined by events with higher probabilities, and lower values of coefficient ($\alpha \to 0$) weight the events more equally, no matter of their probabilities.

The Tsallis and Rènyi entropies were often successfully applied to many diverse practical problems, showing their high usefulness for accurate classification. Among the various proposals the interesting applications concerned, for instance, the work of [28] where the authors applied both entropies for variable selection in computer networks intrusion detection, analyzing models detection capabilities while providing a set of attributes coming from the network traffic. Their results showed that selecting attributes based on the Rènyi and Tsallis entropies can achieve better results as compared to the Shannon entropy.

Some other studies considered image segmentation based on the Tsallis and Rènyi entropies [29]. Their conclusion was that entropic segmentation can give good results but is highly related to an appropriate choice of the entropic index α.

In Ref. [30] the Tsallis and Rènyi entropies applied in C4.5 decision tree have been tested on several high-dimensional microarray datasets. The results showed that use of non-standard entropies may be highly recommended for this kind of data.

3.1. Algorithms for tree generation and pruning

For the purpose of the research the modified C4.5 algorithm for decision tree was used and then both entropies Rènyi and Tsallis were compared. The first algorithm was prepared for growing the tree (Algorithm 1) and the second one for pruning it (Algorithm 2). The modification of the algorithm concerned the pruning part. The algorithm is recursively called so that it works from the bottom of the tree upward, removing or replacing branches to minimize the predicted error on the validation dataset. In order to obtain the optimal split while growing the tree the gain ratio needs to be calculated.

3.2. The other techniques used for comparison

To have a clear reference to the results produced by the trees, logistic regression and two machine learning techniques were also applied. These were artificial neural networks (ANN) and support vector machines (SVM). The choice of the techniques was based on insights that come from literature review, since they are very often used for churn prevention.

Artificial neural networks through their hidden layers and ability to learn seems to be more capable of solving classification problem. Several features of neural networks make them very popular and attractive for practical applications: (1) they possess ability to generalize even if the data are incomplete or noisy; (2) neural nets are non-parametric method which means that they do not require any a priori assumptions about the distribution of the data.

The other method used in our experiments was support vector machines. It is a technique characterized by usage of kernels, absence of local minima, sparseness of the solution and capacity control obtained by acting on the margin, or on number of support vectors. The capacity of the system is controlled by parameters that do not depend on the dimensionality of the feature space. The non-linear function is leaned by linear learning machine which maps inputs into high dimensional kernel induced feature space. SVM are motivated to find and optimize the generalization bounds given for the classification problem using the penalty function for bad classification. They relied on defining so-called epsilon intensive loss function that ignores errors, which are situated within the certain distance of the true value.

```
1  Algorithm: Generate_Tree
   input  : Training samples D, list of attributes L, method for
            attribute selection
   output: Decision Tree
2  Create a node N;
3  if D has the same class C then
4  |  return N as leaf node with class C label;
5  end
6  if if L is empty then
7  |  return N as leaf node with class label that is the most class
   |   in D;
8  end
9  Choose test-attribute α that has the most GainRatio using method
     for attribute selection;
10 Give node N with test-attribute label;
11 Find an optimal split that splits D into subsets D_i (i = 1, ..., k);
12 for i ← 1 to k do
13 |  Add branch in node N to test-attribute = α_i;
14 |  Make partition for sample D_i from samples where
   |   test-attribute = α_i;
15 |  if D_i is empty then attach leaf node with the most class in D;
16 |  else attach node that generate by Generate_Tree (D_i,
   |   attribute-list, test-attribute);
17 end
18 return N
```

Algorithm 1: Growing the tree.

```
1  Algorithm: Prune_Tree
   input  : node with an attached subtree, validation samples W
   output: Pruned Tree
2  leafError = estimated leaf error on W;
3  if node is a leaf then
4  |  leaf error;
5  end
6  else
7  |  subtreeError = ∑_{N_i ∈ children(node)} Prune(N_i);
8  |  branchError = error if replaced with most frequent branch;
9  end
10 if leafError is less than branchError and subtreeError then
11 |  make this node a leaf;
12 |  error = leafError;
13 end
14 else if branchError is less than leafError and subtreeError
     then
15 |  replace this node with the most frequent branch;
16 |  error = branchError;
17 end
18 else
19 |  error = subtreeError;
20 end
21 return error
```

Algorithm 2: Pruning the tree.

Finally, a logistic regression was used for the prediction of the probability of occurrence of an event by fitting the data into a logistic curve. The logistic regression model is used to explain the effects of the explanatory variables in the form of binary response, often using logit transformation, such that we obtain the logistic regression model of the form: $\text{Logit}\{\Pr(Y=1|x)\} = \log\{\Pr(Y=1|x)/(1-\Pr(Y=1|x))\} = \beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+...+\beta_kx_k$, where β_0 is the intercept and β_1, β_2, β_3, and so on are called regression coefficients of x_1, x_2, x_3, respectively. Each of the regression coefficients describes the amount of the contribution. A positive regression coefficient means that

the factor increases the probability of outcome, whilst a negative regression coefficient means that the factor decreases the probability of outcome.

4. The churn dataset

The data used in numerical experiments is a telecom data set (known also as Cell2Cell: The Churn Game) used in the churn tournament in 2003 organized by Duke University and evaluated in [17].

The data set have 71,047 observations and each observation corresponds to the individual customer. There are 78 predictors in it, of which 75 potential variables can be used as input variables to the classification models. The predictors describe a broad set of customer characteristic related to usage behavior (e.g. the average monthly minutes of use over), company interaction (e.g. average number of customer care calls), customer demographics (e.g. the number of children in the household, occupation or marital status of the customer) variables.

TABLE I

The structure of churn dataset.

dataset	No. of observations	Churn rate
training	40,000	50%
testing	31,047	1.96%

All explanatory variables come from the same time period, while binary dependent variable (taking values 0 and 1) labeled as "churn" was observed between 31 and 60 days after explanatory variables. The data set is divided into the learning sample and test sample, having 40,000 and 31,047 observations respectively (Table I). The training sample includes 20,000 of cases classified as churners (labeled "1") and 20,000 of cases classified as non-churners (labeled "0"). The training set is balanced with the churn rate of 50%, which is unrealistic, but it is recommended to overcome the problem of model training when the proportion of churn event is small in the population. The reason for balancing the class distribution in the training sample is to avoid the possibility that the vast majority of the other class may dominate the analysis and make the detection of churn drivers difficult, thus decreasing the predictive accuracy of the model. A discussion on such a procedure can be found in e.g. [31].

In the test sample, which will be used for the model quality assessment, the churn rate is only 1.96% and it refers to actual churn rate observed in the company. Such a small percentage of the one class quite often exists in business practice when building classification models.

5. Analysis and results

5.1. Evaluation measures

To compare the proposed set of machine learning techniques we used two quality measures: area under the ROC curve (AUC) and the lift. The good classifier is characterized by the high value of AUC and high lift.

Since we deal with a problem of binary classification, the model yields two results: positive and negative. There are four possible outcomes: TP — true positives, FP — false positives; also TN — true negatives, and FN — false negatives. In order to construct ROC curve we need to define true positive rate $Tpr = TP/(TP + FN)$ and false positive rate $Fpr = FP/(TP + FN)$.

Defined indicators can be calculated for various values of the decision threshold. The increase of the threshold from 0 to 1 will yield to a series of points (Fpr, Tpr) forming the curve with Tpr on horizontal axis and Fpr on vertical axis as presented in Fig. 1. The curve is named receiver operating characteristics, ROC. The AUC measure is an area under the ROC curve which can be calculated using trapezoidal rule [32]. Theoretically AUC $\in [0; 1]$ and the larger the AUC the closer is the model to the ideal one and the better is its performance.

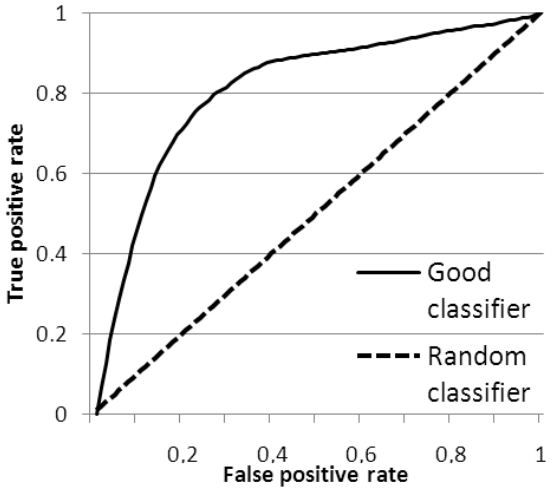

Fig. 1. Receiver operating characteristics curve.

The lift measure is justified by the economic considerations, because the telecom operator does not communicate the retention campaign to a wide customer base, but focuses on a small percentage of approximately 1–2% of the customer group on a monthly basis, characterized by the highest probability of churn. For instance, having the total number of customers of approximately 10 million, a group of 1% of customers is equal to 100 thousand customers per month, which would receive the retention offer. The required input for lift calculation is a dataset that has been "scored" by assigning the estimated churn probability to each case. Next, the churn probabilities are sorted in descending order and for a given customers base, the measure is calculated in the following manner (for the first percentile) [33]:

$$\text{Lift}_{0.01} = \frac{TP_{0.01}}{TP}. \tag{4}$$

The lift measure indicates how precisely we are detecting positive responses (churning customers) in comparison to a random sample of customers.

5.2. Experiments and results

5.2.1. Decision trees

For the purpose of the research the modified C4.5 algorithm for decision tree was used and then both entropies Rènyi and Tsallis were compared. The first algorithm was prepared for growing the tree and the second one for pruning it. The modification of the algorithm concerned the pruning part. The algorithm is recursively called so that it works from the bottom of the tree upward, removing or replacing branches to minimize the predicted error on the validation dataset. In order to obtain the optimal split while growing the tree the gain ratio needs to be calculated.

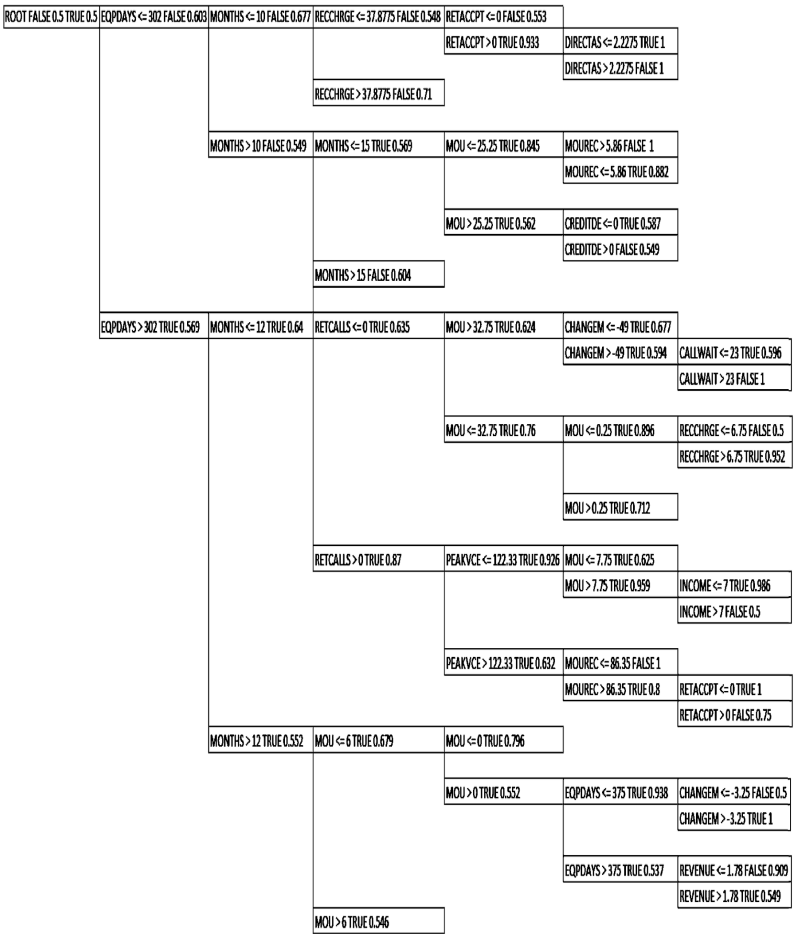

Fig. 2. Decision tree based on Tsallis entropy for $\alpha = 6.5$.

The proposed classification techniques were trained on balanced training sample with churn rate of 50% and then validating the model on the imbalanced testing sample with the churn rate of 1.96%. For decision trees, we considered α starting from 1 to 10 by 0.5. The results obtained on the validation dataset are collected in Table II. The best results and corresponding values of α parameter differ in each case and can be summarized as follows:

1. In general, the Tsallis entropy based trees provided better generalization and the highest lift (3.61) was achieved for the tree with $\alpha = 6.5$ — the structure of the best tree is shown in Fig. 2.
2. The Rènyi entropy based trees obtained slightly worse results and the highest lift (3.28) was achieved for the tree with $\alpha = 2$ — the structure of the best tree is presented in Fig. 3.
3. Both trees resulted in similar AUC performance (60–63%).
4. The Tsallis trees reached the higher lifts for $\alpha \geq 3$ while the Rènyi trees resulted in higher lifts observed only for $\alpha = 2$ and $\alpha = 2.5$. That means, in general, that the trees with the Tsallis entropy are much better in detecting the churning customers in top percentiles of the population (lift = 1 means random ability to distinguish between those two classes).

The Tsallis entropy based tree, as presented in Fig. 2, has 27 leaves on 7 levels (including the root). Each node and each leaf is described by decision rule, class (TRUE if churn was observed and FALSE — otherwise), and the percentage of observations belonging to the majority class.

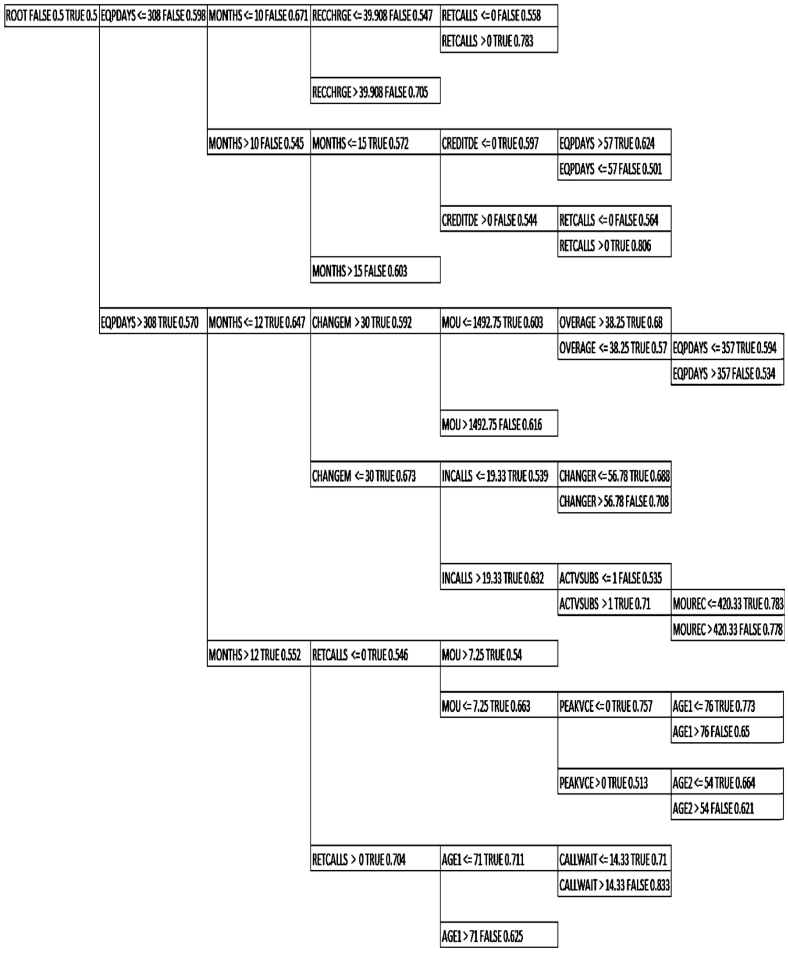

Fig. 3. Decision tree based on Renyi entropy for $\alpha = 2$.

The first variable used for split was EQPDYAS (number of days of the current equipment). If the value of EQPDYAS was greater than 302 then the probability of churn increased, forming a group in which the percentage of churners amounted to 56.9%. On the other levels of the tree it was observed that the following variables were useful for detecting churners: MONTHS (months in service), MOU (mean monthly minutes of use), RETCALLS (number of calls previously made to retention team), RECCHRGE (mean total recurring charge), RETACCPT (number of previous retention offers accepted), PEAKVCE (mean number of in and out peak voice calls), DIRECTAS (mean number of director assisted calls), MOUREC (mean unrounded MOU received voice calls), CHANGEM (% change in minutes of use), CALLWAIT (mean number of call waiting calls), INCOME (customer income), REVENUE (mean monthly revenue).

The Renyi entropy based tree, as presented in Fig. 3, has 39 leaves on 9 levels (including the root), but for the presentation it was pruned to 7 levels.

As previously, the first variable used for split was EQPDYAS but the decision value was 308 this time. If the value of EQPDYAS was greater than 308 then the node consisted of a group in which the percentage of churners amounted to 57%. On the other levels of the tree it was observed that the following variables were useful for detecting churners: MONTHS, MOU, RETCALLS, RECCHRGE, PEAKVCE, MOUREC, CHANGEM, CREDITDE (low credit rating — de), CHANGER(% change in revenues), OVERAGE (mean overage minutes of use), INCALLS (mean number of inbound voice calls), ACTVSUBS (number of active Subs), AGE (age of first household member), AGE2 (age of second household member).

Apart from the Tsallis and Renyi entropies a Shannon entropy tree was built (it is the case for $\alpha = 1$), please see Table III.

5.2.2. Other techniques

Before application of the proposed modeling techniques data preprocessing was applied. It involved Pearson's correlation analysis, and as a result variables with absolute value of correlation coefficient exceeding 0.7 were excluded from the list of potential predictors.

The final logistic regression model was developed using stepwise selection procedure and it includes six variables RETCALLS, CREDITDE, UNIQSUBS (number of unique subscriptions), MOU, ROAM (mean number of roaming calls), EQPDAYS. The variables were significant at $\alpha = 0.05$.

TABLE II

The results of the Tsallis and Renyi trees. The best trees in terms of the lift are presented in bold.

Alpha	Tsallis		Rènyi	
	AUC	Lift	AUC	Lift
1	61.55	2.63	60.83	1.64
1.5	61.95	1.81	61.23	1.64
2	60.62	1.31	**61.13**	**3.28**
2.5	60.86	1.31	61.30	2.79
3	61.32	3.28	62.20	2.46
3.5	61.97	3.12	61.88	3.12
4	61.83	3.45	62.73	1.64
4.5	61.21	2.46	61.34	1.15
5	62.02	2.46	61.29	1.81
5.5	61.17	2.13	63.04	0.99
6	60.77	3.12	62.05	1.15
6.5	**61.17**	**3.61**	63.03	1.64
7	58.74	2.30	62.53	0.99
7.5	61.11	3.28	62.68	1.31
8	61.12	3.45	62.19	1.31
8.5	61.27	2.79	62.46	1.64
9	60.50	2.46	62.34	1.48
9.5	61.22	2.79	62.49	1.48
10	60.84	3.12	59.66	1.31

TABLE III

Summary of the results.

Technique	AUC	Lift
decision tree with Tsallis entropy ($\alpha = 6.5$)	61.17	3.61
decision tree with Rènyi entropy ($\alpha = 6.5$)	61.13	3.28
decision tree with Shannon entropy	62.98	3.24
logistic regression (Logit)	61.65	2.13
neural networks (ANN)	61.40	2.62
support vector machines (SVM)	61.78	2.30

For the neural network training, we used multilayer perceptron (MLP) structure using the back propagation algorithm with a different number of nodes in hidden layer. We started with the MLP model with 72 input variables, 2 neurons in hidden layer and output (MLP 72-2-1) and calculated the evaluation measures. The next step was to add additional neuron to a hidden layer and train MLP 76-3-1 model. We conducted iteratively a number of experiments with each new configuration, retaining the best network in terms of lift and AUC calculated on validation set. The process was stopped when no significant changes in results were observed or the problem with over fitting occurred. The final structure of the model was MLP 72-11-1.

In case of support vector machines, their generalization performance depends on a proper setting of global parameters: C, ϵ and the kernel function. In the experiments, we have arbitrary chosen values of these parameters and tried several different configurations. The final setting of SVM was: (i) 0.01 for parameter ϵ which controls the width of the insensitive zone; (ii) 10 for the capacity coefficient, which determines the trade-off between the model complexity and the degree to which deviations larger than are tolerated in optimization formulation; (iii) and 0.2 for the parameter γ as the kernel function using the radial basis functions. This functions is by far the most popular choice of kernel types, because of their localized and finite responses across the entire range of the real x-axis.

5.2.3. The comparison

The comparison of the results between the trees and other techniques are summarized in Table III. The lifts for all the techniques are depicted in Fig. 4. To calculate the lift, for each model, the population was sorted by the estimated probability of churn in descending order and for a given customers base (decile/percentile), the measure was calculated according to formula (4).

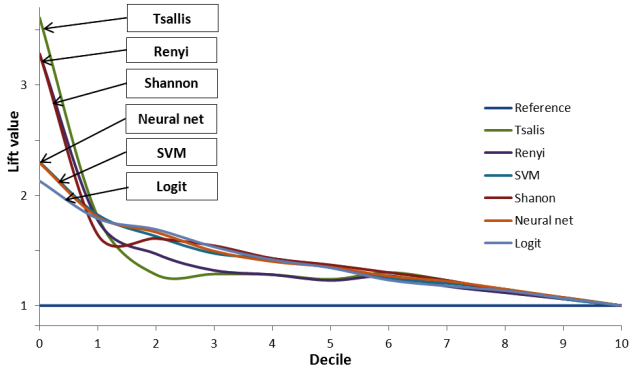

Fig. 4. Lift charts.

The comparison showed that the Tsallis entropy based tree outperformed other decision trees, and also SVM, neural nets and logistic model. The analyzed techniques demonstrated similar value of AUC. The Tsallis entropy based tree reached the highest lift of 3.61 in the first percentile. In Fig. 4, we may observe that the models achieve similar performance in higher deciles of the population. For instance, in the first decile the lift value for all the techniques is very similar — reaching about 1.8. The conclusion is that α-trees can detect precisely the churners in the top percentiles of the population, then for the medium percentiles their predictive power is similar to the other methods.

The last step of evaluation was intended to prepare, for each classification model, a financial simulation on hypothetical retention campaign, assuming the cost of the offer and possible revenues.

Let us assume that telecommunication operator has a total number of active customers of approximately 5 million, then 1% of the customer base is equal to 50 thousand customers per month, which would receive the retention offer. In the analyzed population the churn rate is equal to 1.96%, so it means that 2 customers out of 100 each month are quitting. With a lift of about 3.61, the model gets true positive rate of 7.08% ($3.61 \times 1.96\%$), which translates into 3,540 churner correctly identified by the

model in the first percentile of the population. The communication cost of the retention campaign is estimated on 10 PLN perch each customer, while the revenue from the customer may reach 200 PLN (with the assumption that the relation with customer will last over next 6 months). Then, the net profit of the campaign can reach 208 thousand PLN (see Table IV) for the best model (the Tsallis tree) but also it is possible to generate the loss of 82 thousands PLN for the worst model (Logit).

TABLE IV

Simulation on retention campaign.

	Classification technique					
	Tsallis	Renyi	Shannon	Neural net	SVM	Logit
lift	3.61	3.28	3.24	2.62	2.3	2.13
TP rate (%)	7.08	6.43	6.36	5.14	4.51	4.18
customer base for retention (1%)	50,000	50,000	50,000	50,000	50,000	50,000
no. of churners	3,540	3,217	3,177	2,569	2,255	2,089
cost of the retention offer (10 PLN)	500,000	500,000	500,000	500,000	500,000	500,000
revenue from the customer (200 PLN)	708,000	643,400	635,400	513,800	451,000	417,800
net profit of the campaign (PLN)	208,000	143,400	135,400	13,800	–49,000	–82,200

The results presented in this experiment are encouraging and provide high accuracy of classification, when compared to similar studies on this dataset. For example, the authors in [34] as an assessment of the quality of the model, chose lift in the first decile, which was equal to 2.61 for the best model. Additionally, in our previous study [35] we obtained lift of 3.11 for the first percentile using C&RT tree on the same dataset, while current study delivers improved results.

6. Summary and concluding remarks

We analyzed telecom churn problem using broad set of classification techniques. In particular, we explored the Tsallis and Renyi entropy measures applied to decision trees and compared to other classification methods. We modified a classical decision tree algorithm C4.5 by incorporating parametrized α entropies thus extending classification possibilities. That allowed us to study effectiveness of the trees as a function of the entropy parameter.

Taking into account classification quality we showed that the optimal trees achieve high accuracy of churn detection and they are better classifiers than the other proposed methods — neural networks, support vector machines and logistic regression. In particular, the results can be summarized as follows:

1. The α-based trees outperformed the machine learning methods and logistic regression. The best tree models are considerable good tool to achieve the strategic goals of the company, aimed at churn identification and targeting only a small group of clients with the highest churn probability.

2. We proved that precise selection of entropy measure and its parameters can result in robust models with high ability to predict customer behavior. This was confirmed by the Tsallis and Renyi entropy based trees.

3. The Tsallis entropy based trees are more stable than the Renyi trees — the Tsallis trees reached the higher lifts for $\alpha \geq 3$ while the Renyi trees resulted in higher lifts observed only for $\alpha = 2$ and $\alpha = 2.5$.

4. The α-based trees can detect precisely the churners in the top percentiles of the population, then for the medium percentiles their predictive power is similar to the other methods.

5. When comparing entropy of two distributions, the Shannon entropy assumes implicit tradeoff between contributions from the tails and the main mass of the distribution. In practical applications it is beneficial to control the tradeoff explicitly, as it is important to distinguish weak signal overlapping with the stronger one. With α parameters a large positive value of this measure is more sensitive to events that occur often, while for large negative α it is more sensitive to the events which happen seldom.

To be able to manage the problem the telecom company needs to understand the behavior of customers. The goal is to build up a data-mining model in order to detect pre-churning behavior and allow time to make the right decisions. As it was shown, advanced analytic methods are encouraging and provide high accuracy of classification.

References

[1] K. Gajowniczek, T. Ząbkowski, A. Orłowski, *Ann. Comp. Sci. Inf. Syst.* **5**, 39 (2015).

[2] H. Hwang, T. Jung, E. Suh, *Expert Syst. Appl.* **26**, 181 (2006).

[3] S. Hung, D.C. Yen, H. Wang, *Expert Syst. Appl.* **31**, 515 (2002).

[4] S. Kim, K.S. Shin, K. Park, *Lect. Notes Artif. Int.* **3611**, 636 (2006).

[5] Y. Zhao, B. Li, X. Li, W. Liu, S. Ren *Lect. Notes Artif. Int.* **3584**, 131 (2005).

[6] T. Zabkowski, W. Szczesny, *Expert Syst. Appl.* **39**, 6879 (2012).

[7] G. Nie, W. Rowe, L. Zhang, Y. Tian, Y. Shi, *Expert Syst. Appl.* **38**, 15273 (2011).

[8] A. Keramati, S. Ardabili, *Telecommun. Policy* **35**, 344 (2011).

[9] C. Wei, I. Chiu, *Expert Syst. Appl.* **23**, 103 (2002).

[10] G. Madden, S. Savage, G. Coble-Neal, *Eur. Phys. J. B* **17**, 723 (1999).

[11] J.-H. Ahn, S.-P. Han, Y.-S. Lee, *Telecommun. Policy* **30**, 552 (2006).

[12] J. Burez, D. Van den Poel, *Expert Syst. Appl.* **32**, 277 (2007).

[13] K. Coussement, D. Van den Poel, *Expert Syst. Appl.* **34**, 313 (2008).

[14] D. Seo, C. Ranganathan, Y. Babad, *Telecommun. Policy* **32**, 182 (2010).

[15] M. Owczarczuk, *Expert Syst. Appl.* **37**, 4710 (2010).

[16] B. Huang, M. Kechadi, B. Buckley, *Expert Syst. Appl.* **39**, 1414 (2008).

[17] S. Neslin, S. Gupta, W. Kamakura, J. Lu, C. Mason, *J. Mark. Res.* **43**, 204 (2006).

[18] S. Daskalaki, I. Kopanas, N. Avouris, *Appl. Artif. Intellig.* **20**, 381 (2006).

[19] S. Neslin, *Cell2Cell: The churn game. Cell2Cell Case Notes*, Hanover, NH: Tuck School of Business, Dartmouth College 2002.

[20] C-F. Tsai, Y-H. Lu, *Expert Syst. Appl.* **36**, 12547 (2009).

[21] D. de Waal, J. du Toit, in: *Proc. South Africa Telecommunication Networks and Applications*, Wild Coast Sun 2008.

[22] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, CA 1984.

[23] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA 1993.

[24] C.E. Shannon, *Bell Syst. Tech. J.* **27**, 623 (1948).

[25] C. Tsallis, *J. Stat. Mech. Theor. Exp.* **52**, 479 (1988).

[26] A. Rènyi, in: *Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability*, University of California Press, Berkeley (CA) 1961, p. 547.

[27] R Core Team, *R Foundation for Statistical Computing*, Vienna, Austria 2012.

[28] C.F.L. Lima, F.M. de Assis, C.P. de Souza, *Lect. Notes Artif. Intellig.* **7435**, 492 (2012).

[29] Y. Li, X. Fan, G. Li, in: *Proc. Int. Conf. on Industrial Informatics*, 2006, p. 943.

[30] T. Maszczyk, W. Duch, *Artif. Intellig. Soft Comput.* **5097**, 643 (2008).

[31] B. Donkers, P. Franses, P. Verhoef, *J. Mark. Res.* **40**, 492 (2003).

[32] T. Fawcett, *Mach. Learn.* **31**, 1 (2004).

[33] D.T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley, Hoboken 2014.

[34] I. Bose, X. Chen, *J. Org. Comp. Electron. Com.* **19**, 133 (2009).

[35] K. Gajowniczek, T. Ząbkowski, *Quantitat. Meth. Econ.* **13**, 65 (2012).