Using Machine Learning Methods to Predict Subscriber Churn of a Web-Based Drug Information Platform

Georgios Theodoridis, Athanasios Tsadiras

Using Machine Learning Methods to Predict Subscriber Churn of a Web-based Drug Information Platform

Georgios Theodoridis and Athanasios Tsadiras

Aristotle University of Thessaloniki, Greece
ttgeorgio@csd.auth.gr & tsadiras@econ.auth.gr

Abstract. Nowadays, businesses are highly competitive as most markets are extremely saturated. As a result, customer management is of critical importance to avoid dissatisfaction that leads to customer loss. Thus, predicting customer loss is crucial to efficiently target potential churners and attempt to retain them. By classifying customers as churners and non-churners, customer loss is equated to a binary classification problem. In this paper, a new real-world dataset is used, originating from a popular web-based drug information platform, in order to predict subscriber churn. A number of methods that belong to different machine learning categories (linear, nonlinear, ensemble, neural networks) are constructed, optimized and trained on the subscription data and the results are presented and compared. This study provides a guide for solving churn prediction problems as well as a comparison of various models within the churn prediction context. The findings co-align with the notion that ensemble methods are, in principle, superior whilst every model maintains satisfying results.

Keywords: Machine Learning, Data Mining, Customer Churn, Ensemble Methods, Neural Networks.

1 Introduction to Customer Churn Prediction

Due to saturated markets and intense competition, many companies realize that their existing database is their most valuable asset [1, 12, 22]. This trend is particularly prevalent in subscription services where companies are beginning to move away from traditional, mass marketing strategies in favor of targeted marketing techniques [4]. The idea of identifying churners, meaning customers who are more prone to change and eventually cancel their subscription, has a high strategic priority.

Van Den Poel and Larivi‘ere [6] have suggested that in the current environment, in which potential customers have a huge selection of offers from numerous service providers, attracting new customers is a costly and difficult process. Therefore, more effort to retain existing customers has become necessary for service-oriented companies.

In order to effectively manage customer churn, it is vital to build an effective and accurate customer churn prediction model via multiple predictive modeling techniques. The machine learning models that apply these techniques belong to different categories (linear, nonlinear, ensemble, neural networks) and they vary in theoretical

background, statistical technique, input parameters, number of features selected and included, execution time, and required processing power. Therefore, it is particularly interesting to build, implement and evaluate multiple models with the goal of enriching their bibliographic background within the context of predicting Web-based subscriber churn. In doing so, a methodology guide is presented to assist any Web-based platform in creating and streamlining subscriber retention systems from data collection to model selection and feature analysis. Programmatically, Python 3 was used as it provides excellent modules for data processing (mainly Pandas and NumPy) and method modeling (mainly Sklearn and TensorFlow).

2 Dataset

The dataset used within this paper is a new real-world dataset that was extracted from the usage data of a popular Greek Web-based platform that offers information relating to pharmaceutical products and substances. The platform has 55,000 unique visitors per day and 4,200,000 page views per month. It is a professional tool used by doctors, pharmacists, nurses as well as medical students that aims to support pharmacotherapy decision making. Surface-level information is provided for free whilst users have the option to subscribe in order to access premium analytics and features. In more detail, the dataset reflects 10 years of website usage (July 2010 – December 2020) containing information about the actions/activities that every subscriber performed, accompanied by the general demographic information of each subscriber as well as their subscription history. The original dataset consists of 878,788 activities performed by 779 subscribers throughout 2,270 different subscriptions.

3 Defining the problem & Preprocessing the dataset

In our study, a subscriber is considered churned after not renewing his/her latest subscription within 3 months, as a 3-month plan is the shortest subscription plan offered by the service, hence postponing a renewal for more than 3 months is a direct loss of profit. As a result, the problem of predicting subscriber churn is equated to predicting the non-renewal of subscriptions in the near future. Subsequently, the focus of the problem is moved, from the general subscriber, towards each individual subscription.

At this point, it is important to note the potential differences between classic customer churn and Web-based subscriber churn. After analyzing the renewal delays within the dataset, meaning the time (in days) in between each consecutive subscription, the mean renewal delay is 66 days whilst the median is 6 days, indicating the existence of numerous outliers with large renewal delays (15% are detected as outliers via boxplot analysis). Another crucial observation is that 40% of the subscriptions are 3-month plans (shortest plan provided). Therefore, many users subscribe for the shortest amount of time possible, as they want to use some of the premium features provided by the website, and then do not renew their subscription until they want to use said premium features again. Hence, subscribers oftentimes churn, based on the 3-month churning condition explained in the previous paragraph, but eventually return.

This contrasts the classic customer churn profile observed in mobile telecommunications [8] and insurances [13] where customers sign years-long contracts and, once churned, tend to not return. As such, Web-based subscriber churn should be differentiated from general customer churn for the current dataset as well as any other Web-based subscriber dataset with similar properties.

In order to preprocess the data, we extracted the dependent variable (binary variable containing 0 for retention and 1 for churning) as well as a number of features to be used by the prediction models. In order to extract the dependent variable, every renewal point within the last 5 years is inspected (January 2015 – September 2020) ensuring that the data is up-to-date. The features, described in Table 1, are collected by inspecting the entire dataset in relation to each individual renewal point detected by the aforementioned dependent variable extraction. The features are categorized in two levels; the subscription level (Sn) and the subscriber level (Sr). The subscription level includes information about the latest, at that time, subscription that was about to be renewed or not whilst the subscriber level includes data from every subscription of the related subscriber. For example, when predicting subscriber churn, the number of activities the subscriber made during their latest subscription should be included as a feature (subscription level) but it is also important to include the number of activities the subscriber has historically made throughout their entire subscription history as a separate feature (subscriber level). Using this collection method, an individual timeline per subscriber is formed for every renewal point within the last 5 years, as presented in Fig. 1. The total number of subscription renewal points, which by extension is the sample set to be used for training/testing, is 1238 with the retained-churned ratio being 64.6%-35.4%

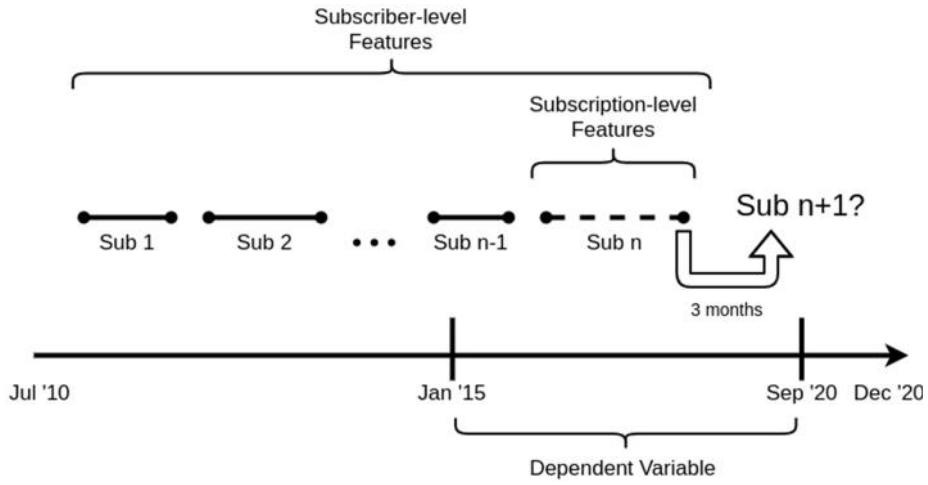

Fig. 1. Presentation of the problem regarding a single potential renewal point (Sub n – Sub n+1) of a subscriber's subscription history, displaying the feature extraction method.

To construct the training and test sets, the sample set is split so that 20% belongs to the test set (248 samples) and the retained-churned ratio within it represents the origi-

nal ratio. The training set is sampled (from the remaining 80%) so that the retained-churned ratio is 50%-50% for better results [14] (700 samples). It is also important to note that every continuous feature is normalized within the closed interval (0, 1) while the categorial features are one-hot encoded with each category becoming a new binary feature. In practice, feature selection was performed to construct the final list of features presented in Table 1 by estimating feature importance (as presented in Section 5.2) as well as feature correlation (using Pearson correlation coefficient [2]) in order to discard potential features that do not correlate with the dependent variable or highly correlate with another feature.

Table 1. List of features and statistics (Mean – Std for normalized continuous features and Yes - No percentages for binary categorial features) of dependent variable.

Feature	Level	Churned	Retained
The subscriber's gender (Are they male or not).	Sr	42% (M)-45%(F)	58% (M) - 55% (F)
The subscriber's age.	Sr	0.49 – 0.13	0.49 – 0.12
The subscriber's city as defined in his profile.	Sr	Varies among 19 categories	Varies among 19 categories
The subscriber's group, found via clustering, performed by the website's developers.	Sr	Varies among 27 categories	Varies among 27 categories
The number of offences the subscriber performed as detected by an internal anti-spam/bot system.	Sr	0.003 – 0.05	0.003 – 0.04
The number of activities the subscriber performed during morning hours.	Sr & Sn	Sr: 0.02 – 0.05 Sn: 0.02 – 0.04	Sr: 0.07 – 0.12 Sn: 0.04 – 0.09
The number of activities the subscriber performed during evening hours.	Sr & Sn	Sr: 0.04 – 0.07 Sn: 0.02 – 0.05	Sr: 0.1 – 0.12 Sn: 0.04 – 0.08
The number of activities the subscriber performed during night hours.	Sr & Sn	Sr: 0.01 – 0.05 Sn: 0.01 – 0.03	Sr: 0.03 – 0.06 Sn: 0.02 – 0.06
The number of drugs the subscriber viewed.	Sr & Sn	Sr: 0.06 – 0.1 Sn: 0.05 – 0.09	Sr: 0.15 – 0.17 Sn: 0.1 – 0.13
The number of drug substances the subscriber viewed.	Sr & Sn	Sr: 0.06 – 0.1 Sn: 0.04 – 0.09	Sr: 0.15 – 0.17 Sn: 0.07 – 0.1
The number of drug packages the subscriber viewed.	Sr & Sn	Sr: 0.05 – 0.09 Sn: 0.05 – 0.1	Sr: 0.13 – 0.16 Sn: 0.11 – 0.14
The number of subscriptions the subscriber made.	Sr	0.03 – 0.06	0.09 – 0.15
This is the first subscription of the subscriber.	Sn	64% – 33%	36% - 67%
The duration (in days) between the start of the latest subscription and the end of the previous one (last renewal delay).	Sn	0.05 – 0.14	0.02 – 0.08
The average of all the last renewal delays.	Sr	0.05 – 0.14	0.02 – 0.06
The last renewal was instant.	Sn	14% - 47%	86% - 53%

The duration (in days) the subscriber is subscribed for. (Cumulatively for Sr level)	Sr & Sn	Sr: 0.13 – 0.15 Sn: 0.09 – 0.11	Sr: 0.25 – 0.21 Sn: 0.10 – 0.12
The duration (in days) of the subscriber's account.	Sr	0.25 – 0.21	0.33 – 0.22
The month the subscription is ending.	Sn	Varies among 12 categories	Varies among 12 categories

4 Categories of Machine Learning Customer Churn Predictive Methods

Predicting customer churn is a binary classification problem outputting two possible classes – churn as the positive class and retention as the negative class. Thus, any machine learning model that is able to handle classification problems is suitable for churn prediction. In the present work, multiple models covering various categories of machine learning methods (linear, nonlinear, ensemble, neural networks) are utilized to compare their suitability and effectiveness. In the following paragraphs a general overview of each model used in this study is presented, summarizing their predictive methods, accompanied by a description of their parameters and their optimized values. Optimization is performed via grid search and 5-fold Cross Validation (CV) so that the Matthews Correlation Coefficient (MCC) metric, which tends to be preferred in binary classification problems over other metrics such as the accuracy, f-score or ROC AUC [7], is maximized.

4.1 Linear Category - Logistic Regression

Regarding the linear category, the popular logistic regression method is studied. Logistic methods expect that the target value is a linear combination of the features [19]. In mathematical notation, if $\hat{y}$ is the predicted value then:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_n x_n$$

In order to optimize the coefficient values w, logistic regression uses a designated solver that attempts to minimize a cost function. For the current problem, the "lbfgs" solver was chosen, alongside with the l_2 cost function, as it is recommended for relatively small datasets and generally considered robust [19].

The method's parameters to optimize, accompanied by the searched values, follow (optimal value in bold):

- C - the regularization term of l_2,Values: 0.1, 0.5, 1, 10, **50**, 100
- The tolerance for stopping criteria, Values: 0.001, **0.00001**, 0.000001

4.2 Nonlinear Category - Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier detection. In a binary classification context,

SVMs try to find a linear optimal hyperplane so that the margin of separation between the positive and the negative samples is maximized [14]. However, in practice, the data is often not linearly separable which necessitates the usage of a kernel function to transform the input space into a higher dimensional feature space. Two kernels will be tested in this paper; the Polynomial kernel and the Radial Basis Function (RBF) kernel that allow nonlinear hyperplanes [23].

Several experiments are performed with various parameters to optimize. The parameters and the searched values are the following (optimal value in bold):
SVM (Polynomial)

- C - the regularization term, Values: 0.1, **0.5**, 1, 10, 50, 100, 250, 500, 750, 1000
- The degree of the polynomial, Values: **2**, 3, 4

SVM (RBF)

- C - the regularization term. Values: 0.1, 0.5, 1, 10, 50, 100, 250, 500, **750**, 1000
- γ – the kernel coefficient. Values: 0.1, 0.01, 0.001, **0.0001**, 0.00001

4.3 Ensemble Methods Category

Random Forest. Random Forests [3] are an ensemble learning method that uses a subset of randomly chosen features to grow decision trees on a bootstrap sample of the training data. After a large number of trees are generated, each tree separately classifies the input. The final class is decided on majority vote amongst the decision trees.

The implementation involves the optimization of a number of parameters. The parameters involved are presented below, followed by the examined values (optimal in bold):

- Function to measure the quality of a split of the decision trees. Values: Information gain or **Gini impurity.**
- The minimum number of samples needed to perform a split. Values: 2, **3**, 4.
- The maximum depth of every decision tree (pre-pruning). Values: 7, 8, 9, 10, **None.**
- The number of randomly chosen features. Values: 9, 10, 11, **12**, 13, 14.
- The number of decision trees to be created. Values: 100, 150, 200, **250**, 300.

XGBoost. XGBoost (eXtreme Gradient Boosting) is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable [5]. It implements parallel tree boosting (also known as GBDT, GBM) under the Gradient Boosting ensemble learning framework. Vinayak and Gilad-Bachrach [20] proposed a new boosting method, the DART booster, to add dropout techniques to boosted trees and reported better results, hence the DART booster will be used for parallel tree boosting.

The parameters to consider optimizing in XGBoost are numerous and a subset of them is chosen for optimization., while the default values will be used [24] for the rest. The searched values follow, having the optimal value in bold:

- The normalization type of the DART booster algorithm. Values: **Decision Tree** or Random Forest.
- The dropout rate of the DART booster algorithm. Values: 0.05, **0.1**, 0.3.
- The learning rate of the booster. Values: 0.01, 0.1, **0.4**, 0.5, 0.7.
- The minimum sum of instance weight (hessian) needed in a child node. Values: 0.5, **1**, 2.
- The minimum loss reduction required to make a further partition on a leaf node of the tree (gamma). Values: **0.005**, 0.001, 0.0005.
- The maximum depth of every decision tree (pre-pruning). Values: 7, 8, **9**, 10, None.
- The number of boosted trees to be created. Values: 100, **150**, 200, 250, 300.

LightGBM. LightGBM is yet another gradient boosting framework that uses tree-based learning algorithms. It is a predecessor to the XGBoost framework and achieves faster training speed and higher efficiency [17]. LightGBM uses histogram-based algorithms [15] which bucket continuous feature values into discrete bins and grows trees leaf-wise (best-first) [21]. Similar to XGBoost, the DART booster will be used.

To optimize LightGBM, the following parameters are examined (the optimal value in bold) while the default values are used for the other parameters [17]:

- The dropout rate of the DART booster algorithm. Values: 0.05, **0.1**, 0.3.
- The learning rate of the booster. Values: 0.01, **0.1**, 0.4, 0.5, 0.7.
- The minimum sum of instance weight (hessian) needed in a child node. Values: 0.5, 1, **2.**
- The maximum depth of every decision tree (pre-pruning). Values: 7, 8, 9, **10**, None.
- The number of leaves in a full tree. Values: 100, 150, **200**, 250, 300.
- The number of boosted trees to be created. Values: 100, 150, 200, **250**, 300.

4.4 Neural Networks

Artificial neural networks are mathematical models inspired by biological neural networks [11]. A neural network is trained by adjusting the weights of each feature, throughout multiple rounds of training called epochs, so that a cost function is minimized or a specific metric is maximized.

The neural network used in this paper is a feed-forward network having the following architecture:

- Batch size is set to 0.25 times the size of the input sample set
- 4 layers
- The rectifier function (ReLU) is used as the activation function of every neuron
- 4 dropout layers, one for each layer

- Optimization via the "Adam" optimizer [9]
- 5000 maximum epochs that are controlled by an Early Stopping function every 250 epochs (terminates the training if the chosen metric, MCC, is not further optimized within 250 epochs and restores the optimal feature weights)

To optimize the parameters of the neural network, a 50 trial Bayesian Optimization [10] is used, instead of grid search, to fasten the process as the number of all possible combinations is extremely high. The parameters that are optimized are (optimal value in bold):

- The number of neurons per layer. Values: 50 – 1000, step 50. **Optimal:** layer 1 set to 600 neurons, 2 to 50, 3 to 250 and 4 to 250.
- The dropout rate of every dropout layer. Values: 0, 0.3, 0.5. **Optimal:** layer 1 set to 0.3, 2 to 0.5, 3 to 0.3 and 4 to 0.
- The learning rate of the optimizer. Values: 0.1, **0.01**, 0.001.

5 Results

5.1 Training & Testing

Having optimized the parameters, the models can now be trained and tested. In reality, using 5-fold Cross Validation (CV) is, in and of itself, a training and testing procedure performed five times within the training set. Therefore, tracking metrics while performing cross validation can measure the effectiveness of a model. In Table 2, the average MCC as well as the average ROC AUC of the optimal parameters is tracked, accompanied by the results (tp-true positives, tn-true negatives, fp-false positives, fn-false negatives) of the trained models classifying the test set as well as the training set itself. The Cohen's Kappa [18] of the test set is also presented to measure the accuracy of each model in relation to the random classifier (based on class frequency) to better understand the usefulness of every model on imbalanced data. The ROC Curves of every model on the test set are also depicted in Fig. 2.

Table 2. Model metrics and results.

Method (rank)/ Metrics	LR (7)	SVM Pol (6)	SVM RBF (5)	RF (2)	XGB (1)	LGBM (3)	NN (4)
CV MCC	0.34	0.33	0.35	0.42	**0.44**	0.38	0.40
CV ROC AUC	0.72	0.72	0.73	**0.78**	0.77	0.76	0.77
Training Set tp	264	242	231	**350**	**350**	**350**	261
Training Set tn	246	251	255	**350**	**350**	**350**	264
Training Set fp	104	99	95	**0**	**0**	**0**	86
Training Set fn	86	108	119	**0**	**0**	**0**	89
Test Set tp	63	59	58	65	**68**	64	64
Test Set tn	96	113	115	112	114	**116**	110
Test Set fp	64	47	45	48	46	**44**	50
Test Set fn	25	29	30	23	**20**	24	24

Test Set Precision	0.5	0.56	0.56	0.58	**0.6**	0.59	0.56
Test Set Recall	0.72	0.67	0.66	0.74	**0.77**	0.73	0.73
Test Set Kappa	0.29	0.36	0.36	0.41	**0.45**	0.43	0.39

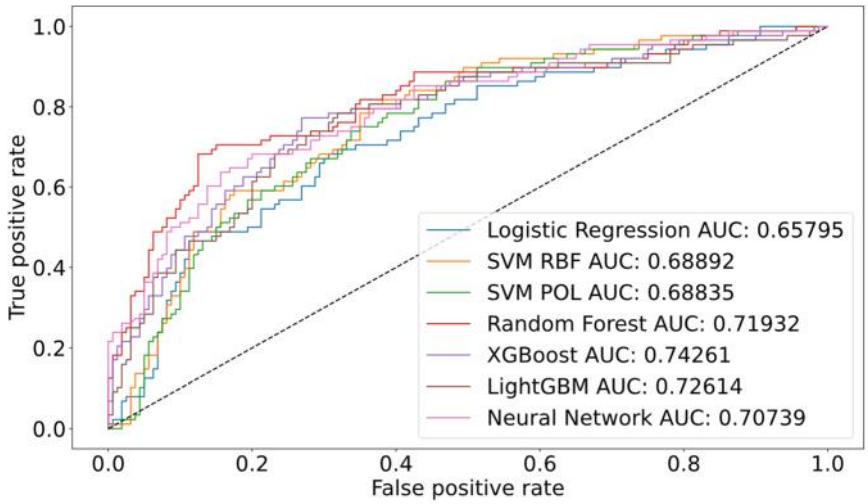

Fig. 2. ROC Curves and AUC Scores of every model on the test set.

The results, both in Table 2 and Fig. 2, rank logistic regression as the weakest of competitors and the ensemble methods as the better choices. LightGBM seems to fall behind XGBoost by a slim margin and the Neural Network is outperforming the SVM models whilst its CV metrics manage to reach those of LightGBM. Detecting churners seems to be challenging but the results are satisfactory. Relatively high recall reassures the prediction of most churners but low precision indicates the existence of numerous false alarms (non-churners classified as churners) amongst them. Nonetheless, Kappa values greater than 0.4 suggest moderate agreement with the truth deeming all ensemble methods practically useful.

Furthermore, the predictive models output not only the predicted class but also the probability of churning. Using said probability, a cumulative gain chart is created by sorting every prediction from highest to lowest. Cumulative gain charts display the accumulated percentage of positive and negative classes on the percentage of sorted samples. Such analysis is highly important as approaching every single potential churner, in an attempt to retain them, is oftentimes impossible due to time restrains or financial limitations. For example, in Fig. 3 and by using the Random Forest classifier, if the top 20% of possible churners are targeted for retention, 45% of the overall churners will successfully be approached but, simultaneously, 25% of the overall non-churners will be pointlessly targeted. The chart indicates that Random Forest is the superior model in this type of analysis thus deeming it more practical in real life scenarios. It also indicates that numerous non-churners are strongly classified as churners, suggesting that some subscribers seemingly act like churners but ultimately renew

their subscriptions. This observation is not necessarily negative; subscribers that use the service less (most important indicator of churning as suggested by the Feature Importance analysis in the next section) will renew their subscription nonetheless, potentially proposing that they are loyal and they depend on the service even if they rarely take advantage of it.

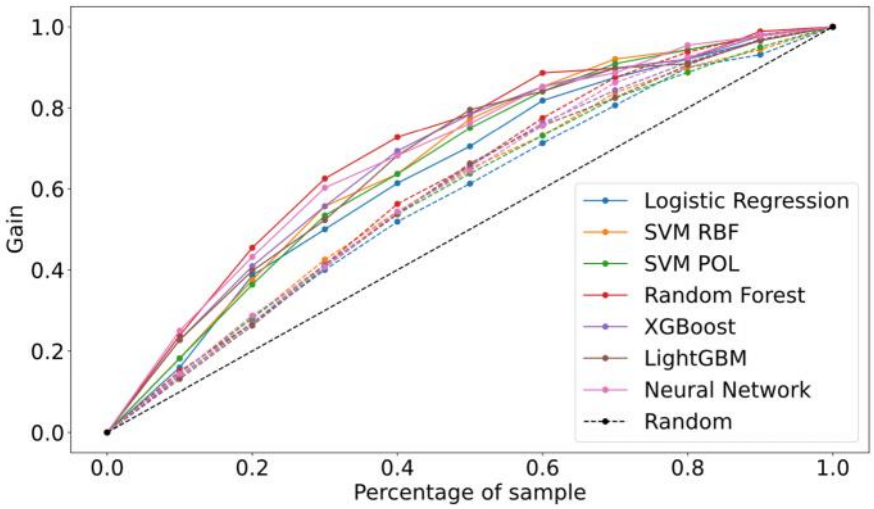

Fig. 3. Cumulative Gain Chart of every model (Solid lines - Positive Class, Dashed lines - Negative Class)

5.2 Feature Importance

In this section, an overview is given regarding the features that play the most important role on the prediction. The outcome of the Random Forest importance measures shall be used for two reasons: (i) Neural Networks are “black box” models and, consequently, extremely tedious to analyze for feature importance, (ii) compared to other ensemble methods, Random Forest is the only one that samples random features which makes it really effective against multicollinear features [16].

In Table 3 the top 10 most impactful features are presented. It becomes clear that the website usage is, almost singlehandedly, the prime indicator of churning or retention. The subscriber level features also seem to outweigh their subscription counterparts. The only features amongst the top 10 that are not related to usage are a) the cumulative time subscribed, indicating to having loyal customers that choose long lasting subscription plans, and b) the average renewal time, hinting to the conclusion that “if a subscriber is, on average, skeptical or forgetful about renewing their subscription then they will continue being skeptical or forgetful about it (the same applies for loyal subscribers but on the opposite way)”. It should be noted that one-hot encoded features might show low impact count in isolation but the original categorial feature might be of relative greater importance necessitating further analysis on that regard to extract a better estimate.

Table 3. Top 10 important features.

Feature	Level	Average Importance per Prediction
Number of evening activities	Sr	8%
Number of drug packages viewed	Sr	8%
Number of morning activities	Sr	7%
Number of drugs viewed	Sr	6%
Number of drug substances viewed	Sr	6%
Number of nighttime activities	Sr	5%
Average renewal time.	Sr	5%
Cumulative time subscribed	Sr	5%
Number of drugs viewed	Sn	4%
Number of morning activities.	Sn	4%

6 Conclusions & Future Research

In this study, various methods of different machine learning categories are presented to predict customer churn and assist a web-based drug information platform to sustain preexisting customers in a saturated market. Every model is overviewed, optimized via 5-fold Cross Validation and compared to one another. The results indicate that, even though every model delivers respectable results, the ensemble methods are the strongest choice. Moreover, the results point to unpredictable non-churners that hinder the accuracy of every model, but subscriber loyalty is also existent and prevalent.

Future research will focus on introducing new methods as well as further optimizing the neural network, mainly its architecture, which can vastly change by removing static parts and introducing them as parameters, as well as escaping from the monotonic feed-forward design. Data-wise, upscaling methods could be applied so that the training set can remain within class balance without wasting samples. Lastly, we anticipate the practical usage of this work by the development team that trustingly provided us the dataset and the potential metadata we can collect from it to further optimize all models.

Acknowledgements. The authors would like to thank Ergobyte Informatics S.A. for providing the dataset and for their valuable comments and suggestions on the preparation of this study.

References

1. Athanassopoulos, A. D.: Customer satisfaction cues to support market segmentation and explain switching behavior. Journal of Business Research, 47(3), 191–207 (2000).
2. Benesty, Jacob, et al.: Pearson correlation coefficient. Noise reduction in speech processing. Springer, Berlin, Heidelberg, 1-4 (2009)
3. Breiman, L.: Random forests. Machine Learning, 45(1), 5–32 (2001).

4. Burez, J., Van den Poel, D.: CRM at Canal+ Belgique: Reducing customer attrition through targeted marketing. Expert Systems with Applications (2007).
5. Chen, Tianqi., Guestrin, Carlos.: XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794 (2016).
6. D. V. den Poel, B. Larivi‘ere.: Customer attrition analysis for financial services using proportional hazard models. European Journal of Operational Research, 157(1):196–217 (2004).
7. Davide Chicco, Giuseppe Jurman.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. Chicco and Jurman BMC Genomics (2020).
8. D. G. M. Mozer, R. Wolniewicz, H. Kaushansky.: Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. IEEE Transactions on Neural Networks, 11, 690–696 (2000).
9. Diederik P. Kingma, Jimmy Lei Ba.: ADAM: A method for stochastic optimization. Published as a conference paper at ICLR 2015 (2015).
10. Dillon Sterling, Tyler Sterling, YuMing Zhang, Heping Chen.: Welding Parameter Optimization Based on Gaussian Process Regression Bayesian Optimization Algorithm. IEEE International Conference on Automation Science and Engineering (CASE) Aug 24-28, 2015. Gothenburg, Sweden (2015).
11. Gupta, Neha.: Artificial neural network. Network and Complex Systems 3.1, 24-28 (2013).
12. Jones, M. A., Mothersbaugh, D. L., Beatty, S. E.: Switching barriers and repurchase intentions in services. Journal of Retailing, 76(2), 259–374 (2000).
13. K. Morik, H. Kopcke.: Analysing customer churn in insurance data a case study. In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 325–336, New York, USA (2004).
14. Kristof Coussement, Dirk Van den Poel.: Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. Expert Systems with Applications 34 313–327 (2008).
15. Li, Ping, Qiang Wu, Christopher J. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. Advances in Neural Information Processing Systems 20 (2008).
16. Lidia Auret, Chris Aldrich.: Empirical comparison of tree ensemble variable importance measures. Chemometrics and Intelligent Laboratory Systems, Volume 105, Issue 2, Pages 157-170 (2011).
17. LightGBM’s documentation, https://lightgbm.readthedocs.io/, last accessed 2021/10/3
18. McHugh, Mary L.: Interrater reliability: the kappa statistic. Biochemia Medica, vol. 22, no. 3, 276-282 (2012) Available: https://hrcak.srce.hr/89395, last accessed 2021/1/5.
19. Pedregosa et al.: Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830 (2011).
20. Rashmi, K. V., Gilad-Bachrach, R.: DART: Dropouts meet Multiple Additive Regression Trees, http://arxiv.org/abs/1505.01866 (2015).
21. Shi, Haijian.: Best-first decision tree learning. The University of Waikato, (2007).
22. Thomas, J. S.: A methodology for linking customer acquisition to customer retention. Journal of Marketing Research, 38(2), 262–268 (2001).
23. Vert, JP., Tsuda, K., Schölkopf, B.: A Primer on Kernel Methods. Kernel Methods in Computational Biology. MIT Press. 35-70 (2004).
24. XGBoost’s documentation, https://xgboost.readthedocs.io/, last accessed 2021/10/3.