A FRAMEWORK BASED ON OPEN-SOURCE TECHNOLOGIES FOR AUTOMATED CHURN PREDICTION IN NON-CONTRACTUAL BUSINESS SETTINGS

Milan Mirković[3]
Stevan Milisavljević[4]
Danijela Gračanin[5]

https://doi.org/10.31410/itema.2018.6

Abstract: *Predicting customer churn has become increasingly important for companies competing in contemporary markets, as modern technologies keep tipping the scales of power and influence into the hands of customers. Hence, devising and executing retention campaigns targeting the population that is at the risk of being "lost" can make a big difference to the financial performance of a company. In this paper, we present a framework based on open-source technologies that makes evaluation of different churn definitions in a non-contractual business setting easy, in terms of resulting model performance. In particular, we propose an automated approach to feature engineering, model creation, model evaluation and model selection that should enable companies to quickly assess the effects of choosing a particular interval of inactivity as a churn definition period on the potential value of planned retention activities.*

Keywords: *Churn prediction, machine learning, framework, automation, non-contractual business setting*

1. INTRODUCTION

Technological advances have made it easy and convenient for consumers not only to make purchases anytime and anywhere, but also to compare products and services offered by different companies and make a switch from one supplier to another at the whim. This makes the competition for every single buyer more fierce than ever, and puts a lot of stress on corporate Customer Relationship Management (CRM) to keep existing customers happy and loyal – as obtaining new customers that might churn quickly is often more expensive than retaining existing customers [1]. Since buyers invariably leave at some point in their lifetime, it has become increasingly important for companies to identify those prone to leaving as soon as possible so appropriate retention activities can be taken – in hope that they can be persuaded to remain a customer. Positive effects of such campaigns can really make a difference to the financial performance of the company [2], thus it is essential to try and keep the retention rate as high as possible.

Customer churn prediction is a relatively well-explored domain that has been in the focus of researchers and industry practitioners for quite a while. However, most of the efforts have been aimed at predicting churn in industry branches where businesses operate under a subscription model, such as telecommunications [3] - [4], banking [5] or energy sector [6]. These contractual

[3] Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
[4] Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
[5] Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia

settings make churn flags explicit, i.e. the moment when customers churn is known precisely (e.g. they cancel their subscription or it expires on a particular date). In non-contractual business settings, such as retail, the moment of churn is not explicitly defined, as customers may make purchases at their leisure – hence a business understanding (i.e. definition) of churn has to be derived.

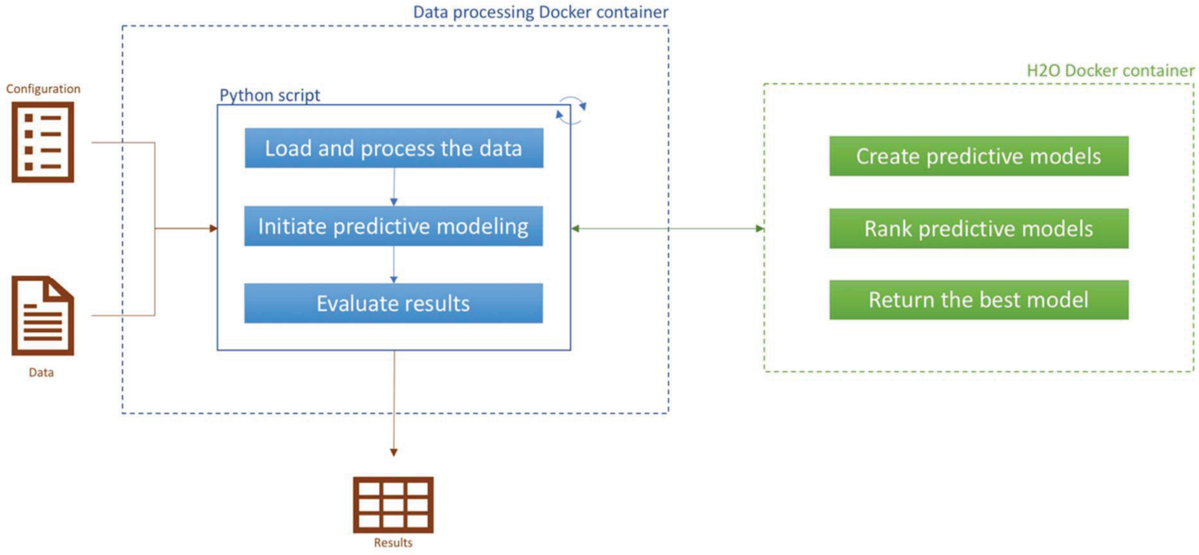
One way to do this is to define a particular period of customer inactivity (i.e. they do not make a purchase for three successive months) and then deduce from the transactional data whether a customer has churned at a particular point in time or not. Naturally, this raises the question of how to come up with a good definition (i.e. inactivity period to use for churn) and how useful the resulting models that use that definition are from a business perspective.

2. FRAMEWORK

The framework we propose in this paper treats churn prediction as a binary classification problem and allows finding a predictive model that maximizes the desired metric not only by testing the performance of different machine learning algorithms, but by manipulating the input data as well. In doing this, it sheds some light on what might be reasonable values to use for churn definition, derived from the data itself.

Two distinct Docker [7] containers are run within the framework, as shown in Figure 1. One contains the Python code that, given a set of configuration parameters (churn definition period and the number of months for calculating monthly features differences), loads the transactions data and processes it (i.e. creates features and churn labels from the raw data). It then sends the processed data to the other container – which contains a running H2O [8] instance – for predictive modeling. H2O container creates multiple predictive models based on the incoming dataset, ranks them and returns the results of the top performing one. The entire process is then repeated for the next set of configuration parameters until all are exhausted.

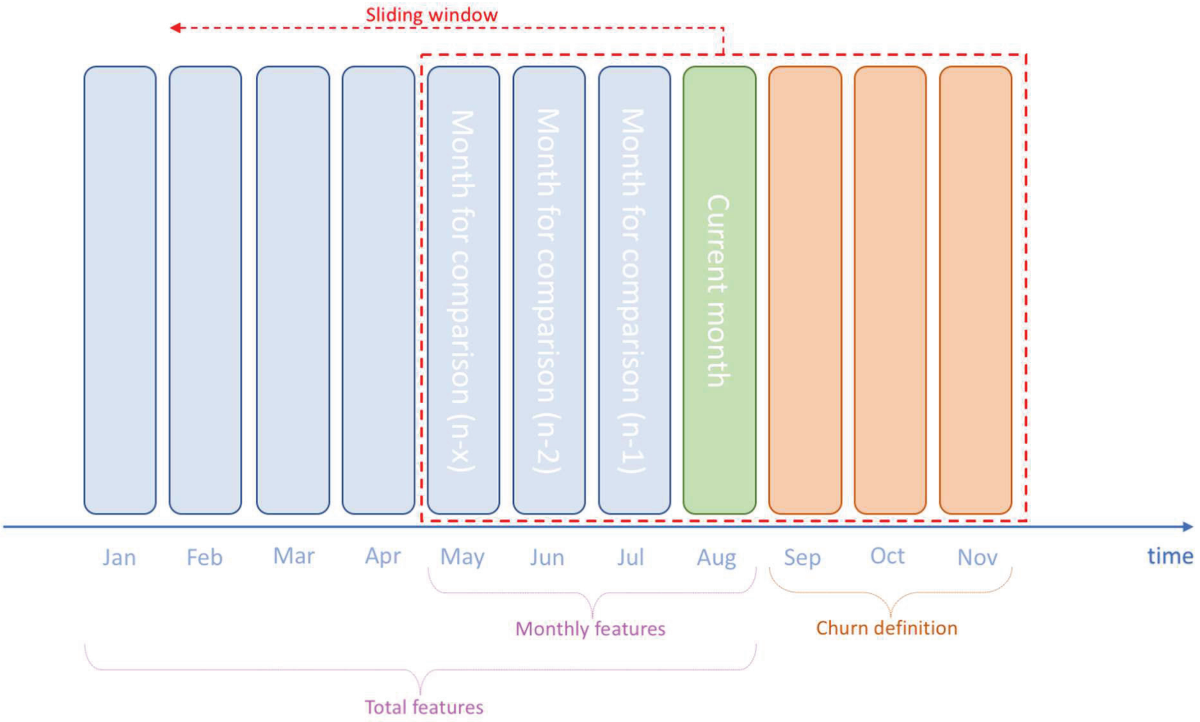
Figure 1: Framework for automated churn prediction



2.1. DATA PROCESSING

To create the dataset that is submitted to H2O for predictive modeling, a sliding-window approach is used. Window width is defined by two user-configurable parameters: churn definition period and the number of months to use for calculating differences between monthly features. In addition, cumulative features over the entire customer lifetime are added as Total features. This is illustrated in Figure 2.

Figure 2: Sliding-window approach to feature extraction



Only simple Recency, Frequency and Monetary (RFM) features are used for predictive modeling, as they require no additional information to raw transactions data (which is expected to be on the invoice-line level). They are calculated on two separate levels: monthly aggregates and totals. Months are used as units for aggregation as they present a natural cutoff point for summarization in various business processes (e.g. customers might be billed monthly) as well as for potential retention activities. Base features derived are: number of items purchased (per month and total), number of distinct items purchased (per month and total), number of purchases made (per month and total), amount spent (per month and total), average line amount (per month and total), average amount per purchase (per month and total) and days since last purchase (relative to the end of month). Additional features calculated are differences in values of base features on monthly level between a particular month and the *n*-th month prior to it (e.g. amount spent in March versus amount spent in February).

Churn flags are derived by using a certain period of customer inactivity (i.e. not making a purchase) in successive months (e.g. customer did not make a purchase in three successive months).

To keep model evaluation as close to the real-world conditions (i.e. least biased) as possible temporal splits are used to create the training and test sets, by using the latest month of processed data as the test set and all the ones prior to it as the training set.

2.2 PREDICTIVE MODELING

H2O provides automation options for the machine learning workflow, which include training and tuning of many models within a user-specified time-limit. Stacked ensembles are automatically trained on collections of individual models to produce highly predictive ensemble models – which can be ranked and the best one selected for future data scoring.

Presented framework utilizes this functionality by leveraging H2O's AutoML Interface that is passed the training data for creating predictive models and test data for ranking them. Area under receiver operating characteristics curve (AUC) is used as the main metric to determine the best performing model, as it is often used as a measure of quality of a probabilistic classifier and is close to the perception of classification quality that most people have [9].

Besides AUC, gains-charts are also constructed for every top-performing model, since they are a useful tool for evaluating the business impact of putting a particular model into production.

2.3. WORKFLOW

To provide an estimate on what might be a good value to use as a churn definition period given a transactions dataset, the framework requires two aforementioned input parameters (the maximum length of possible churn definition period and the maximum number of months for calculating differences between monthly features). Given these, two vectors containing integers ranging between one (1) and the respective parameter value are constructed, and a grid search is performed on their cartesian product (where the search criteria is the model with the highest AUC). This yields a model with the best performance with respect to the observed metric that at the same time uncovers the potential churn definition to be adopted.

Of course, care should be taken that plausible values that make business sense are passed as input parameters to the framework, as otherwise the outcome of the process might not be actionable (e.g. it might turn out that one year as a churn definition period yields the best model, but the model might actually be useless in the real world as one year could be simply too broad of a definition from the business perspective). In addition, to evaluate the potential business impact of the resulting models, gains charts should be inspected and used in conjunction with existing domain-knowledge to select the appropriate model for planned retention activities.

3. EXPERIMENT

To test the framework, we used a publicly available dataset [10] that comprises 541,909 lines present in customer transactions for a UK-based and registered non-store online retailer. Dataset spans the period between the 1st December 2010 and 9th December 2011. Each line is described by 8 variables: Invoice number, Stock code, Description, Quantity, Invoice Date, Unit price, Customer ID and Country. After filtering (removing rows with missing values and rows where quantity purchased is negative – as these are most likely product returns) and removing the Description and Country columns (as they are not used in any calculations), the dataset comprised 397,884 lines and 6 columns.

We set the framework input parameters to (3, 3), meaning that there were 9 different models constructed (using: [1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1], [3, 2] and [3, 3] as parameter values across different runs). Ranking of models (according to the AUC obtained) is presented in Table 1.

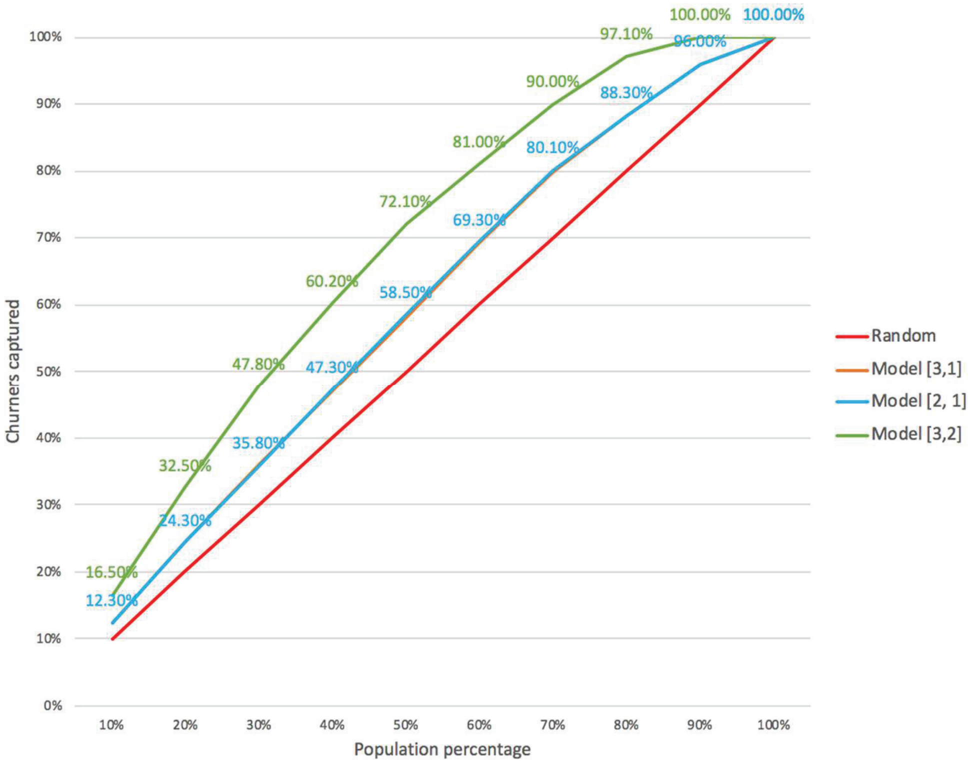
Table 1: Model ranking for the sample dataset

Model parameters ***[months for calculating differences, churn definition period]***	***AUC***
[3, 1]	0.909
[2, 1]	0.894
[3, 2]	0.891
[1, 1]	0.865
[2, 2]	0.849
[1, 2]	0.807
[3, 3]	0.792
[2, 3]	0.787
[1, 3]	0.785

According to the AUC values obtained, one or two months could be good candidates for churn definition period, and two or three months for calculating differences between monthly features might be used.

However, to see how effective would the models be if they were put into production (i.e. used to drive some retention activities, such as campaigns with special offers), gains chart was constructed for the top three performing models, and is shown in Figure 3.

Figure 3: Gains chart for the top three models



The value on the horizontal axis represents the percentage of customers to be targeted (e.g. by a campaign) while the value on the vertical axis shows the percentage of churners that would be captured by addressing the respective percentage of customers. If no model was used to make predictions on who is likely to churn, the expectation would be that a proportional percentage of churners is captured within the respective percentage of targeted customers (e.g. if a campaign was aimed at 20% of the population, it would be expected that 20% of all churners are captured within it). This expectation is labeled as “Random”.

The gains chart obtained indicates that the [3, 2] model has the highest potential business value, as it consistently captures the highest percentage of churners within every decile of the target population, when compared to other models.

4. CONCLUSIONS

Besides empowering buyers, contemporary technologies also make it possible for businesses to analyze their customers and use data-driven insights to come up with solutions that help them tailor the products and services they offer to the end-user’s liking. In this paper we presented a framework that is based on open-source technologies and that allows companies to identify not only customers who are likely to churn but to come up with plausible churn definition periods from the transactional data as well.

Experimental results show potential for this framework to be used in real-world settings, especially when scalability and extensibility are in question; the choice of technologies makes parallelization and scaling-up easy (Docker / H2O), while at the same time additional data sources can be plugged-in without too much effort (which should lead to more accurate predictive models).

REFERENCES

[1] Reinartz, W. J., Kumar, V. (2003). The Impact of Customer Relationship Characteristics on Profitable Lifetime Duration. *Journal of Marketing*, *67*, pp. 77–99.

[2] Van den Poel, D., Lariviere, B. (2004). Customer Attrition Analysis For Financial Services Using Proportional Hazard Models. *European Journal of Operational Research*, *157*, pp. 196–217

[3] Nath, S. V, Behara, R. S. (2003). Customer Churn Analysis in the Wireless Industry: A Data Mining Approach. *Proceedings-Annual Meeting of the Decision Sciences Institute*, (561), pp. 505–510.

[4] Huang, B. Q., Kechadi, T. M., Buckley, B., Kiernan, G., Keogh, E., Rashid, T. (2010). A new feature set with new window techniques for customer churn prediction in land-line telecommunications. *Expert Systems with Applications*, *37*(5), pp. 3657–3665.

[5] Dudyala Anil, K., Ravi, V. (2008). Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, pp. 4–28.

[6] Pribil, J., & Polejova, M. (2017). A Churn Analysis Using Data Mining Techniques: Case of Electricity Distribution Company. In *Proceedings of the World Congress on Engineering and Computer Science*, Vol. I, pp. 1–6, San Francisco, USA.

[7] Merkel, D., (2014). Docker: lightweight Linux containers for consistent development and deployment, *Linux Journal*, 2014.

[8] H2O.ai (2018), Open source software, https://www.h2o.ai/ (accessed: 28.10.2018).

[9] Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, pp. 4626–4636.

[10] Chen, D., Sain, S. L., Guo, K. (2012). Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management,* pp. 197–208.