European Journal of Operational Research 295 (2021) 664–674

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Analytics, Computational Intelligence and Information Management

How training on multiple time slices improves performance in churn prediction

Theresa Gattermann-Itschert*, Ulrich W. Thonemann

Department of Supply Chain Management and Management Science, Albertus-Magnus-Platz, University of Cologne, Cologne 50932, Germany

ARTICLE INFO

Article history:
Received 19 October 2020
Accepted 21 May 2021
Available online 29 May 2021

Keywords:
Analytics
Churn prediction
Marketing
Retailing
Machine learning

ABSTRACT

Customer churn prediction models using machine learning classification have been developed predominantly by training and testing on one time slice of data. We train models on multiple time slices of data and refer to this approach as multi-slicing. Our results show that given the same time frame of data, multi-slicing significantly improves churn prediction performance compared to training on the entire data set as one time slice. We demonstrate that besides an increased training set size, the improvement is driven by training on samples from different time slices. For data from a convenience wholesaler, we show that multi-slicing addresses the rarity of churn samples and the risk of overfitting to the distinctive situation in a single training time slice. Multi-slicing makes a model more generalizable, which is particularly relevant whenever conditions change or fluctuate over time. We also discuss how to choose the number of time slices.

1. Introduction

In industries with high customer acquisition costs, customer retention is a crucial success factor (Reichheld, 1996; Zeithaml, Berry, & Parasuraman, 1996). For targeting customers with retention activities, companies have to identify customers who might churn. Machine learning classification models have been used for churn prediction tasks in different industries. Well known examples of such industries are telecommunications (Coussement, Lessmann, & Verstraeten, 2017; Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012), banking (Kumar & Ravi, 2008; Larivière & Van den Poel, 2005), and TV/newspaper subscription (Ballings & Van den Poel, 2012; Burez & Van den Poel, 2007; Coussement & Van den Poel, 2009). By training on a large sample of customers, models learn to identify customers who are likely to stay with the company and those who are likely to churn.

Most churn prediction studies consider a single time window of data and train and test their models on a single time slice. By doing so, they potentially miss extracting relevant information from the data and cannot properly capture that conditions and drivers of churn change over time. Changing environments make it difficult to train a model that upholds its performance in the future (Risselada, Verhoef, & Bijmolt, 2010).

In this study, we structure data into time slices and use multiple time slices for training a model (multi-slicing). Each time slice considers data from a specific time window. The features and labels are computed relative to a reference point within that window. Multiple time slices are then combined to one training set. With this concept, a model can be trained on samples from different points in time.

Churn prediction studies that use multiple time periods within the training data are rare. Gür Ali and Arıtürk (2014) show for a case in the banking sector that the technique leads to better results than the common approach of training on only one slice of data. However, they do not analyze the drivers of performance improvement. We develop and compare different multi-slicing versions to identify these drivers.

Our key research questions are whether multi-slicing makes trained models more generalizable in a changing environment, which role training on samples from different time slices plays, and what effect the number of time slices has on performance.

To answer these questions, we introduce multiple versions of single- and multi-slicing and compare predictive performance while varying the given time frame of data and the number of time slices. We facilitate the comparison by suggesting a notation of time slices. We also derive how to use the multi-slicing approach in combination with cross-validation to allow for statistical testing.

Using data from a large European convenience wholesaler, we benchmark three classification techniques and use the best performing one, random forests, to conduct our analyses regarding multi-slicing. We find that training on multiple time slices in-

* Corresponding author.
E-mail addresses: theresa.gattermann@uni-koeln.de (T. Gattermann-Itschert), ulrich.thonemann@uni-koeln.de (U.W. Thonemann).

https://doi.org/10.1016/j.ejor.2021.05.035
0377-2217/

creases predictive performance. We show that multi-slicing exploits a given time frame of data more efficiently than using a single time slice for training. We analyze different versions of multi-slicing to isolate the effects contributing to performance improvement. We find that models trained with multi-slicing gain generalization capability even if the training set size is not increased. This indicates that there is inherent value in training on observations from multiple time slices. For our dataset, using more time slices in the training data enhances performance with the first additional time slices resulting in the highest improvement.

2. Literature review

2.1. Churn prediction with classification techniques

Verbeke, Martens, Mues, and Baesens (2011) and De Caigny, Coussement, and De Bock (2018) provide an overview on literature in churn prediction modeling. Benchmarking studies on classification algorithms (Verbeke et al., 2012), data preparation techniques (Coussement et al., 2017), and methods to handle class imbalance (Burez & Van den Poel, 2008; Zhu, Baesens, Backiel, & Vanden Broucke, 2017) show the variety of models that have been developed to predict churn successfully. For insights into the time to churn for different groups of customers or different types of churn, alternate methods like survival analysis can be applied (Burez & Van den Poel, 2008; Larivière & Van Den Poel, 2004). We focus on classification algorithms to predict the customers that are likely to churn in the near future.

Logistic regression, support vector machines, and random forests are among the most common classification techniques and frequently used as benchmark classifiers (De Caigny et al., 2018; García, Nebot, & Vellido, 2017).

Logistic regression is well-studied and typically serves as a benchmark in churn prediction studies (Coussement et al., 2017). It is a statistical technique fitting a linear model to separate samples into two classes. Due to its low complexity, run times are low and interpretability is good. At the same time, complex relationships are difficult to detect and interactions between features are difficult to consider.

Support vector machines (SVMs) aim for good generalization capability by maximizing the margin of samples to the decision boundary (Vapnik, 1995). Several studies have focused on SVMs in churn prediction (e.g. Coussement & Van den Poel, 2008; Kumar & Ravi, 2008; Lessmann & Voß, 2009) or used it as a benchmarking technique (Verbeke et al., 2011; Zhu et al., 2017).

Random forests are an ensemble technique, creating multiple decision trees and giving the most frequent class decision as output. With bagging (bootstrap aggregating), the method can avoid overfitting by averaging over the opinion of many different trees built on bootstrap samples (Breiman, 2001). Various churn prediction studies indicate robust and good predictive performance of random forests (Buckinx & Van den Poel, 2005; Burez & Van den Poel, 2009; Coussement & Van den Poel, 2008; 2009; Verbeke et al., 2012).

In churn prediction, class imbalance naturally occurs because churners are usually less frequent than non-churners. To deal with absolute and relative rarity of the minority class, sampling and cost-sensitive learning are the most popular approaches (Weiss, 2004). Under-sampling discards samples from the majority class and over-sampling duplicates or constructs samples from the minority class to create a more balanced dataset. Synthetic Minority Oversampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) is a popular example. Sampling methods change the class distribution and sample size, whereas cost-sensitive learning deals with imbalance on an algorithm level by placing more weight on the minority class. Weiss (2004) recommends to use cost-sensitive learning because it does not discard valuable information and has been shown to perform better than under- and over-sampling (Japkowicz & Stephen, 2002). An example for a cost-sensitive learner are weighted random forests (Chen, Liaw, & Brieman, 2004). They have been found to outperform classic random forests in churn prediction (Burez & Van den Poel, 2009).

2.2. Model testing approaches

Churn prediction model performance can be evaluated with two different testing approaches: Out-of-sample testing evaluates a model on the same period it is trained on, splitting the available customer base into a subset used for training and a subset used for testing. Out-of-period testing evaluates a model on a more recent period than it is trained on, assessing model performance beyond the training time frame on data reserved as a hold-out test set.

If market demand, pricing, customer behavior, competitive behavior and other factors that are relevant for customer churn are stable over time, out-of-sample and out-of-period testing yield similar results. If the environment and underlying drivers for churn change over time, an accurate assessment of predictive performance requires out-of-period testing.

While out-of-sample testing is still the prevalent method in churn prediction literature, there are several studies that use out-of-period testing. Zahavi and Levin (1997) and Neslin, Gupta, Kamakura, Lu, and Mason (2006) argue that out-of-period testing is necessary to assess how well a model performs on new data beyond the training period. Wei and Chiu (2002) show that the predictive power of their model on telecommunications data is time-sensitive and already declines after one period. Burez and Van den Poel (2008) report similar results for pay-TV customer churn. For internet service provider and insurance data, Risselada et al. (2010) attribute the decrease in model performance over time to changing market conditions. Out-of-period testing has been applied more frequently in recent studies, such as Glady, Baesens, and Croux (2009) (financial services), Chen, Fan, and Sun (2012) (various datasets), Coussement and De Bock (2013) (online gambling) and Óskarsdóttir et al. (2017) (telecommunications).

2.3. Model training with data from multiple periods

Only few studies consider multiple time periods within the training data. Gür Ali and Arıtürk (2014) are one exception. They suggest a sampling framework that uses multiple samples per customer observed at different points in time instead of the common approach with one sample per customer observed at one fixed point in time. The framework is applied in the banking sector and the authors show that using multiple period training data can improve predictive performance over using only one period of training data. However, the analysis is restricted to one example with a fixed number of periods. The effect of varying the training set size and the number of included periods on predictive performance is not investigated.

Two recent studies have applied an approach similar to Gür Ali and Arıtürk (2014). Seppälä and Thuy (2018) predict churn for housing loan customers and Leung and Chung (2020) for retail bank customers. Both provide indications that training on multiple time slices improves predictive performance.

The multi-slicing approach has shown promising results for specific applications, but the approach has not been solidly analyzed, such that, for instance, the role of the training set size and the number of time slices is unclear. It has not yet been investigated whether performance improvement can be maintained even when the training set size is independent of the number of time slices, but samples come from multiple time slices. This is relevant

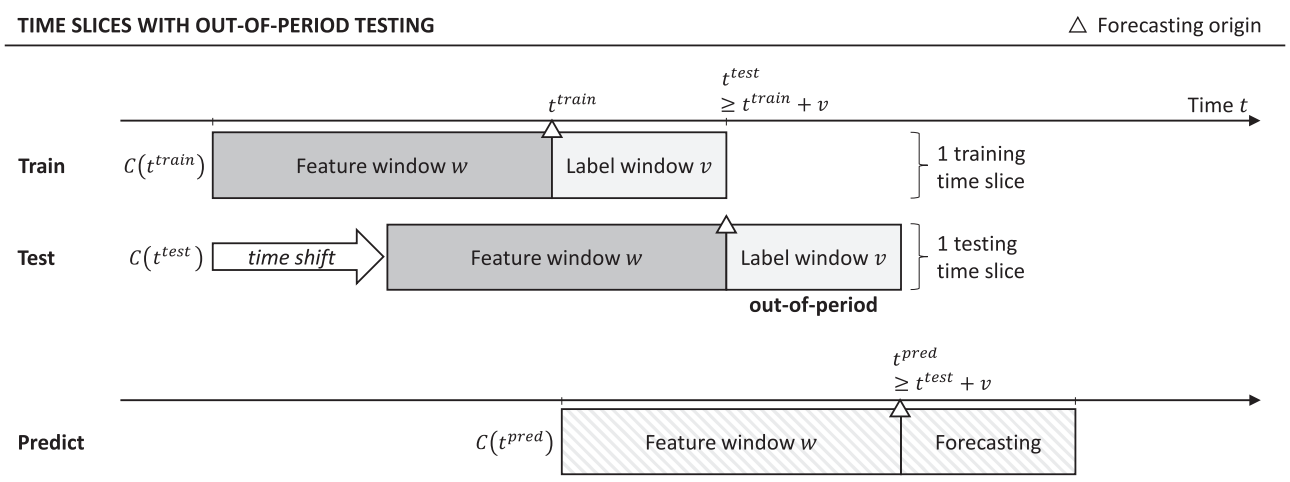

Fig. 1. Time slice concept illustrated with out-of-period testing.

for understanding the effects which contribute to performance enhancement. It is also unclear, how predictive performance is affected by the number of time slices used for training.

3. Methodology

For multi-slicing, transactional time-stamped data must be processed so that multiple time slices can be used for training. In the following, we introduce our notation for time slices and illustrate the concept of out-of-period testing (Section 3.1) before elaborating on the multi-slicing approach (Section 3.2).

3.1. Time slices

A time slice is an extract from the available data for a specific time frame. It contains both feature and label windows, with the forecasting origin t in between as a reference point.

Relative to t, a time window w of previous data is used for calculating features on customer behavior during this time. A time window v of subsequent data is used for label creation regarding the churn of customers. The customer base of a time slice is the set $C(t)$ of active customers at the forecasting origin t. For these customers, features and corresponding labels are computed.

The feature matrix X_t consists of non-time-varying features and time-varying features per customer at time t. Non-time-varying features are typically customer master data like region, shop type, or payment method. Time-varying features are generated from events during the time window w up to forecasting origin t such as invoices, deliveries, and customer relationship management (CRM) actions. For each customer $i \in C(t)$ considered at time t, a label y_{it} is created based on the transactional data during the following time window v. The label y_{it} is 1, if the behavior of customer i during v indicates churn, and 0 otherwise.

The sample size varies over time depending on the number of active customers $|C(t)|$. Different observations of the same customer can appear in different time slices.

We visualize the time slice concept in Fig. 1 by showing the different time slices used for training, testing, and prediction. We distinguish between the forecasting origins t^{train}, t^{test}, and t^{pred}.

One time slice is used for training the model with features derived from window w and labels from window v. A more recent time slice is used for testing, ensuring non-overlapping label windows. The testing time slice is shifted by v so that the forecasting origin of the test set is $t^{test} \geq t^{train} + v$. This results in testing on an out-of-period label window. The figure also shows that forecasting a future not yet seen can happen at the earliest at $t^{pred} \geq t^{test} + v$, after the known data has been used for training and testing.

3.2. Multi-slicing

Structuring available transactional data into time slices opens up the possibility of multi-slicing. With this concept multiple time slices can be generated and combined to one training set.

With multi-slicing, customers are viewed at different times t_k^{train}. The features and labels of each time slice $k \in \{1, \ldots, K\}$ are a snapshot of the customers at the forecasting origin t_k^{train}. The most recent slice views customers at t_K^{train} and the oldest slice views customers at t_1^{train} shifted backwards by $K-1$ periods. The training data set is constructed by a stacked feature matrix $X_{K,t^{train}} = [X_{t_1^{train}} \cdots X_{t_K^{train}}]^\top$ and a corresponding stacked label vector. $X_{K,t^{train}}$ consists of up to $\sum_k |C(t_k^{train})|$ rows. Note that $K-1$ additional periods of data are used for training on K time slices instead of only on one time slice.

Fig. 2 illustrates that the training set for multi-slicing consists of multiple time slices. The time slices required for testing and potential prediction are also included.

Our multi-slicing approach is neither an under-sampling technique nor an over-sampling technique. Instead, multi-slicing accesses more actual observations from historical data of both the minority and majority class without changing the underlying class distribution. Creating time slices generates more samples and addresses the absolute rarity or lack of data issue while only moderately expanding the time horizon used. We thereby follow the recommendation by Weiss (2004) to not discard any data, but to use all information available. Multi-slicing facilitates learning on past churners and non-churners in addition to the most recent ones.

3.3. Cross-validation with time slices

For statistically testing our results, we derive a technique for conducting cross-validation with out-of-period testing while using multiple time slices within the training set. We consider two-fold cross-validation. The technique entails partitioning the dataset into two equally sized subsets or folds which are each used once for testing.

In each iteration j of cross-validation, it is essential that the subsets of customers used for training $C_j(t^{train})$ and testing $C_j(t^{test})$ are not overlapping. It is not straightforward how to ensure this when applying out-of-period testing since different observations of the same customers can appear in both training and testing time

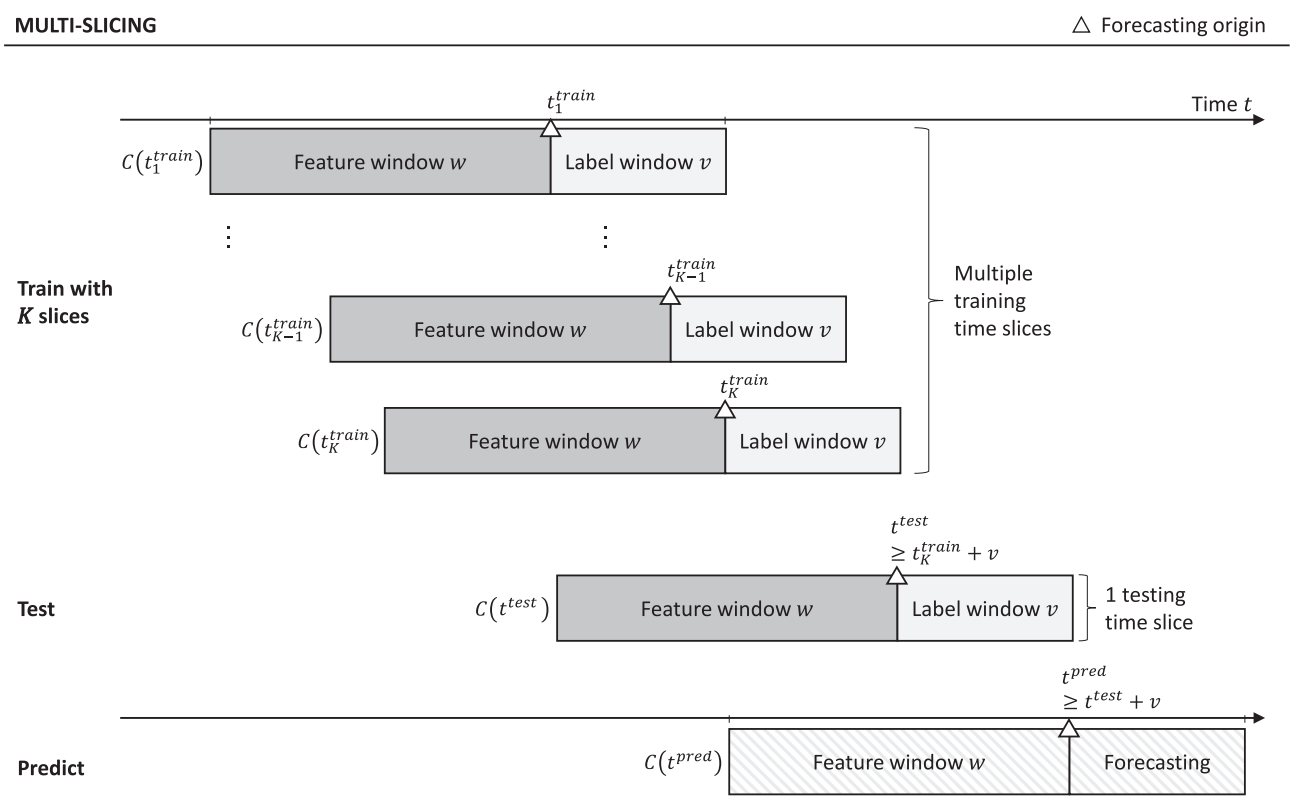

Fig. 2. Training on multiple time slices.

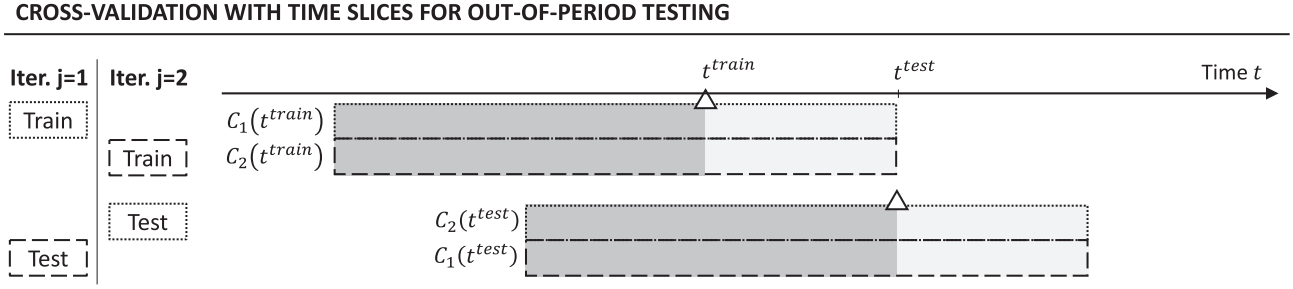

Fig. 3. Deriving folds for 2-fold cross-validation with time slices for performing out-of-period testing.

slice. We illustrate our approach for two-fold cross-validation in Fig. 3.

The goal is to use each observation only once for testing, either in iteration 1 or in iteration 2 and to never train on customers that are also used for testing in the same iteration.

The testing slice customers are assigned to two equally sized disjoint testing folds $C_1(t^{test})$ and $C_2(t^{test})$. The training slice customers are also divided into subsets, ensuring that customers used for testing in the same iteration are not included in the corresponding training subset. Customers in $C_1(t^{test})$ that also appear in the (older) training slice are assigned to the opposite fold for training $C_2(t^{train})$ and vice versa. Customers that only appear in the training slice are randomly assigned to the two folds.

In the first iteration, a model is trained on $C_1(t^{train})$ and tested on $C_1(t^{test})$ and in the second iteration, it is trained on $C_2(t^{train})$ and tested on $C_2(t^{test})$. The subset of customers trained on and the subset of customers tested on are disjoint in each iteration:

$C_1(t^{train}) \cap C_1(t^{test}) = \varnothing$ and $C_2(t^{train}) \cap C_2(t^{test}) = \varnothing$.

We extend this to multi-slicing so that for each customer included in a testing fold, all previous observations of the customer are excluded from the corresponding training fold (see Fig. 4). Customers that only appear in the training set are randomly assigned to one of the two training folds. The approach generates disjoint subsets of customers for training and testing in each iteration:

$\left(C_1(t_1^{train}) \cup \cdots \cup C_1(t_{K-1}^{train}) \cup C_1(t_K^{train})\right) \cap C_1(t^{test}) = \varnothing$ and

$\left(C_2(t_1^{train}) \cup \cdots \cup C_2(t_{K-1}^{train}) \cup C_2(t_K^{train})\right) \cap C_2(t^{test}) = \varnothing$.

4. Empirical study

We develop a churn prediction model for one of Europe's largest convenience wholesalers (company) that sells beverages, tobacco, food, and other essential supplies to small convenience retail stores (customers). These business-to-business (B2B) customers can chose how frequently and which products they buy from the company. Customers can end the relationship without canceling a contract with the wholesaler. Defining churn in a non-contractual setting such as ours is difficult. The company experiences customer churn in terms of a substantial drop in revenue, known as partial defection (Buckinx & Van den Poel, 2005).

In our study, the dataset contains around 5000 active B2B customers. For feature creation, we use customer master data as well as data from transactional databases such as invoicing, delivery, and CRM. We include RFM variables (recency, frequency, and monetary value) and features specific to the industry.

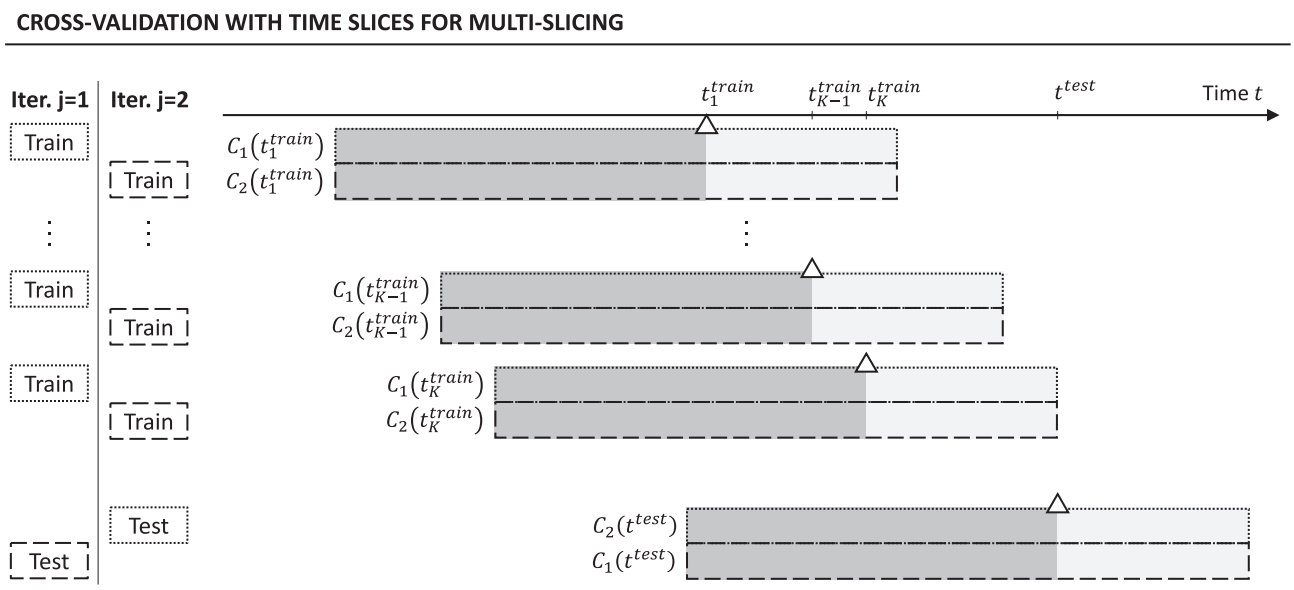

Fig. 4. Deriving folds for 2-fold cross-validation with multi-slicing.

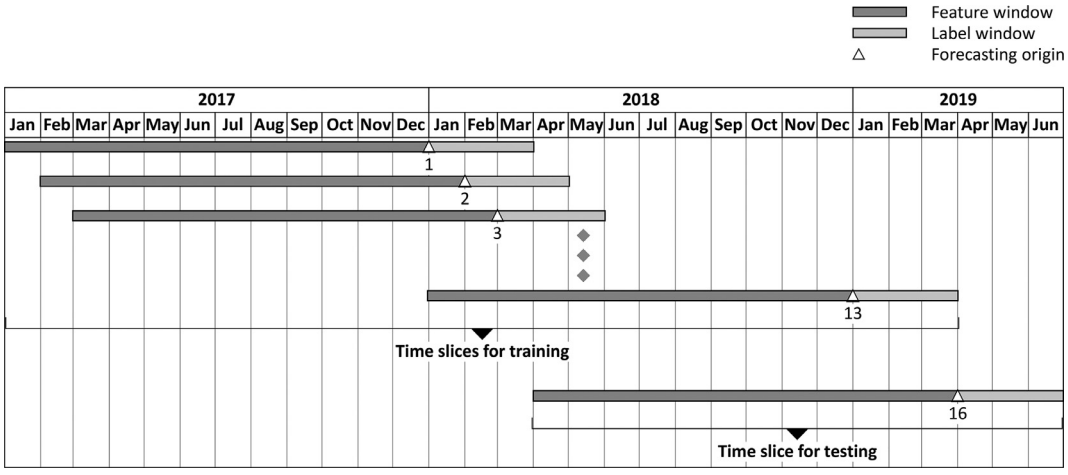

Fig. 5. Available data horizon with time slices for training and testing.

4.1. Dataset

Our dataset spans 2.5 years (January 2017–June 2019). We analyze the loyal customer base, which is of most interest to the company. The company considers a customer loyal, if he or she has been active for each of the previous 12 months and has a mean interpurchase time of at most two weeks. The customer base varies between 4945 and 5463 customers over time. We use a time period for feature computation of $w = 12$ months and $v = 3$ for labels. The earliest time slice has January 1st, 2018 as forecasting origin. Segmenting the data into multiple time slices, each shifted by one month, results in 16 possible forecasting origins to be used for training or testing (see Fig. 5). Reserving the latest time slice for testing and ensuring that the label windows of training and testing are not overlapping leads to a training set that can encompass at most 13 time slices.

4.2. Churn definition

Churn must be defined to generate labels for the classification task. In a non-contractual setting such as ours, defining churn is not straightforward, because churn is not revealed by the termination of a contract. Customers do not necessarily churn abruptly at one distinctive point in time but often gradually move purchases to competitors. Since partial defection is a likely behavior of customers that will eventually churn completely, Buckinx and Van den Poel (2005) argue that it should be the prediction focus. Miguéis, Van den Poel, Camanho, and Falcão e Cunha (2012) and Miguéis, Camanho, and Falcão e Cunha (2013) use substantial decrease in purchase volumes as churn indicator.

In our setting, we rely on expert domain knowledge from the company to identify partial defectors among the loyal customers, which have been regularly active in each of the last 12 months. At the company, customers are considered churners, if their three-month spending fulfills two conditions: first, their spending decreases by at least 50% in any product segment compared to the previous three months and second, their total three-month spending decreases by at least 30% compared to the previous three months.

Fig. 6 indicates that the churn rate of the company fluctuates around 10% and exhibits seasonality. A varying level of churn suggests that conditions and behavior patterns also change over time, making out-of-period testing with a time shift between training and test set essential. It is also a reason to train on multiple time slices, because then conditions from different time slices are

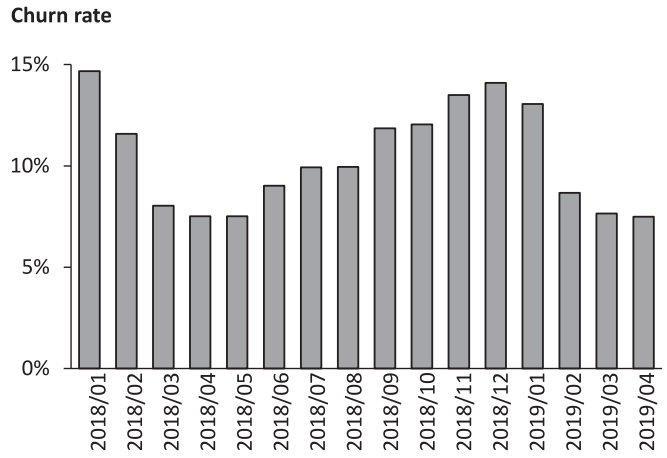

Fig. 6. Seasonality of monthly churn rate.

included and overfitting to one particular point in time can be prevented.

4.3. Features

In close consultation with business experts from the company, we created features that are potential predictors of churn. The features can be grouped by their data source.

Invoice data Recency, frequency, and monetary value (RFM) have been shown to be good churn predictors (Bose & Chen, 2009; Buckinx & Van den Poel, 2005; Miguéis et al., 2013; Tamaddoni Jahromi, Stakhovych, & Ewing, 2014). We calculate the RFM features based on invoicing data from the 12 months before the forecasting origin. Values are normalized by customer specific monthly or quarterly means, which facilitates comparing customers and capturing trends. Volatility of values can be measured by the coefficient of variation (CV). For interpurchase times (IPT), for instance, the CV is distributed differently for customers who stayed and customers who churned (Fig. 7). The invoice data also reveals the number of product segments ordered from as well as credits, debits, and returns.

Delivery data Both missing a communicated delivery time window and failing to deliver the full quantity of products ordered is considered poor delivery performance. We include both types of delivery failures separately in our features as well as on-time in-full (OTIF) as a combined measure. To enable comparability between customers, we divide the days with missed delivery windows by days ordered and the product value not delivered by total revenue of the customer. Chen, Hu, and Hsieh (2015) have shown for the case of a logistics company with B2B relationships, that the number of delivery failures divided by the total number of transactions is a valuable churn predictor.

CRM data The company's CRM system records contacts between sales agents and customers. We include the number of contacts via phone or in person and the days since the last contact as features that quantify the interactions between the company and the customers.

Customer master data It has been found that the continuity of a customer relationship increases with its length (Anderson & Weitz, 1989). We include the length of the relationship and other features that the company believes are relevant for churn prediction such as region, store type or basic price level.

We refer to Appendix A for a full list of feature types used and the variants created. Logistic regression and SVMs require features to vary on a similar scale, so rescaling preprocessing steps are required. We standardize the continuous features so that they have a mean of zero and unit variance in the training set. This preprocessing step is not necessary for random forests.

5. Experimental setting

We conduct our analyses with a churn prediction model using the most relevant features and the best performing of the three benchmark classification techniques regularized logistic regression, linear SVM, and random forests on our dataset. We perform nested cross-validation with an inner loop to tune hyperparameters and an outer loop to report performance results. We assess classification performance with two evaluation metrics.

To determine the benefit of the multi-slicing technique, we compare it to different versions of training on a single time slice. For insights into the effect of sample size, we evaluate a downsized version of multi-slicing. We analyze the impact of the number of time slices by varying the slices included in the training set.

5.1. Feature selection

For feature selection, we run the classifiers with recursive feature elimination (RFE). A wrapper approach like RFE has been shown to be superior to filter techniques (Kohavi & John, 1997; Somol, Baesens, Pudil, & Vanthienen, 2005). RFE was introduced for SVM (Guyon, Weston, Barnhill, & Vapnik, 2002) and has been applied in the churn prediction domain by Lessmann and Voß (2009) and Farquad, Ravi, and Raju (2014). Using backward elimination, RFE recursively removes the least important features and

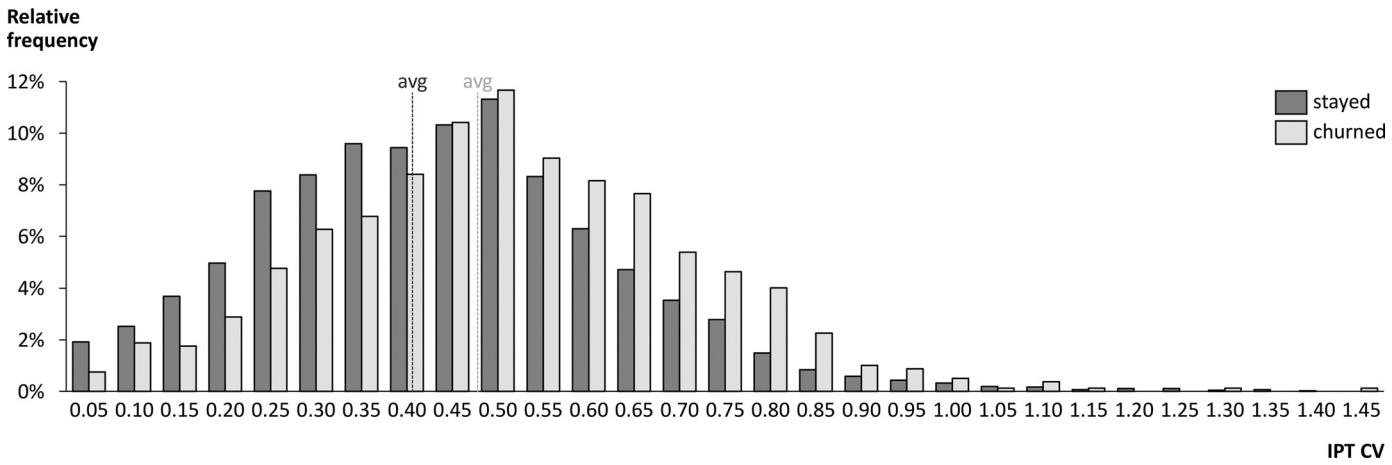

Fig. 7. Coefficient of variation of interpurchase times (IPT CV): Relative frequency of values by class.

can be applied with different classifiers on the basis of coefficients or feature importance scores. Similarly to Verbeke et al. (2012), we determine the number of features to choose with the wrapper approach for each classifier by evaluating the predictive performance (see Appendix B).

5.2. Evaluation metrics

We determine classifier performance with two metrics that are established in the churn prediction domain: the area under the receiver operating curve (AUC) and top-decile lift (TDL) (Coussement et al., 2017; De Bock & Van Den Poel, 2012; De Caigny et al., 2018; Verbeke et al., 2012). Both are based on the confusion matrix. It measures for classification tasks with binary labels (1: positives = churn, 0: negatives = no churn) how many predictions were correct (true positives (TP) and true negatives (TN)) and how many were incorrect (false positives (FP) and false negatives (FN)). The distribution of samples in the confusion matrix depends on the probability threshold of the classifier decision function. The following threshold-dependent measures can be calculated:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

The receiver operating curve (ROC) considers the trade-off between these two measures, with an increasing TPR coming at the cost of a higher FPR. The area under the curve provides a summarized performance measure across thresholds that can be used to compare models (Egan, 1975) and is calculated by approximating

$$\text{AUC} = \int_0^1 \text{TPR}\, d\text{FPR}.$$

The resulting AUC value indicates how well the model is capable of distinguishing between churners and non-churners. It ranges between 0 and 1 and can be interpreted as the probability that a randomly chosen churner has a higher score than a randomly chosen non-churner. A baseline model which assigns the class by chance therefore has an AUC score of 0.5.

We also report TDL, another common measure that is especially relevant for practitioners who use a churn prediction model for selecting the top customers to target with retention activities. For this purpose the performance in the top decile is more important than the overall model capability. TDL evaluates how much better a model is at identifying churners among the top decile compared to randomly targeting customers. This is determined by assessing

$$\text{Lift} = \frac{\frac{TP}{TP+FP}}{\beta}$$

for the top decile, dividing the precision among the top 10% customers with the highest churn probabilities by the overall churn rate β. A TDL of 1 means, that the model has no benefit over a baseline model which targets a random sample of customers. The maximum value achievable is $\frac{1}{\beta}$ (Piatetsky-Shapiro & Masand, 1999).

5.3. Model training

We compare regularized logistic regression, linear SVM, and random forests to select the best performing technique for further analyses. For tuning the hyperparameters of each classification method considered, we conduct cross-validation with five folds within the training data. We use stratification to allocate samples from the two classes evenly to the validation sets. We apply grid search to iterate over combinations of reasonable values for the parameters (see Table 1) and select the best performing combination scored by AUC.

Table 1
Hyperparameters and ranges used during grid search.

Classifier	Hyperparameter	Values
Logistic regression	penalty	L2 regularization
	solver	liblinear
	regularization C	$[10^{-5}, 10^{-4}, \ldots, 10^{2}]$
	class weight	balanced
SVM	kernel	linear
	regularization C	$[0.2, 0.3, \ldots, 1.2]$
	class weight	balanced
Random Forests	max. number of features	$[\sqrt{F}, 2\sqrt{F}, 3\sqrt{F}, \min(F, 4\sqrt{F})]$
	max. tree depth	$[3, \ldots, \min(F, 15)]$
	min. samples per leaf	2
	class weight	balanced

For logistic regression with L2 regularization, which penalizes the squared feature weights, we test different values for the regularization parameter. For a basic version of SVMs with a linear kernel, we also vary the regularization parameter during grid search.

We control the complexity of our random forest model by varying the 'maximum number of features' to consider when searching for the next best split and the 'maximum tree depth'. For both parameters, low values decrease the risk of overfitting. Since we also apply RFE, we use suitable ranges depending on the number of features F for grid search. To further reduce the risk of overfitting, we require at least two samples per leaf.

Our dataset exhibits a high class imbalance with only around 10% of customers being churners. Since we analyze the impact of varying the training set size with multiple time slices on predictive performance, we do not use sampling methods that change this size such as under- or over-sampling but instead apply cost-sensitive versions of the classifiers. We use balanced class weights which are inversely proportional to the class frequency and thereby assign more weight to samples from the minority class. This implicitly leads to treating both classes as if they had similar sizes without changing the sample size.

5.4. Cross-validation and statistical testing

We apply nested cross-validation. While the hyperparameters are tuned with an inner cross-validated grid search, we report out-of-period test results over an outer loop. We assess AUC and TDL performance over five replications of twofold cross-validation (5x2-fold cv) (Dietterich, 1998). This experimental set-up is popular in the churn prediction domain because it allows for statistical testing when comparing algorithms, testing methods or sampling techniques (e.g. Ballings & Van Den Poel, 2015; Burez & Van den Poel, 2009; Coussement & De Bock, 2013).

In each iteration of two-fold cross-validation in the outer loop, we ensure that the customer subsets used for training and testing are non-overlapping. In Section 3.3, we describe how this is maintained when applying out-of-period testing and when training on multiple time slices within the training data.

Results are analyzed for significant differences by pairwise Wilcoxon signed-rank tests with the Bonferroni correction for multiple comparisons (Demšar, 2006; Wilcoxon, 1945).

5.5. Single- and multi-slicing versions

We conduct experiments with different versions of training on a single time slice (single-slicing) and on multiple time slices (multi-slicing). To assess the benefit of the multi-slicing technique, we compare it to different versions of training on a single time slice. For simple single-slicing, we use a 15 months time window ($w = 12$ months and $v = 3$ months). Multi-slicing with time slices of the same length requires a longer time frame of data. To eval-

uate which method exploits a given time frame better, we make both methods comparable by extending the feature window of single-slicing accordingly (extended single-slicing).

For example, a training time window of 18 months has three additional months of transactional data that can be used to either extend the feature creation window ($w' = 15$ months) or to create multiple time slices. When shifting each time slice by one month, in total $k = 4$ time slices can be created out of 18 months of training data.

To gain insight into the drivers of multi-slicing performance, we compare multi-slicing to a downsized version. Samples are randomly drawn from the multiple time slices available, but the training set size is limited to the number of samples in the single-slicing approach. We also prevent including more than one observation of customers that appear in multiple time slices to exclude any effects this might have on performance. We refer to this approach as downsized multi-slicing.

We use the following versions of single- and multi-slicing in our experiments:

- *Simple single-slicing* as a baseline using only 12 months as feature window
- *Extended single-slicing* using the maximum feature window size possible with the given months of training data
- *Multi-slicing* using as many time slices as possible, each with a feature window of 12 months and shifted by one month
- *Downsized multi-slicing* using samples drawn from multiple time slices, downsized to sample size of latest training slice

6. Results

We first select the best performing of the three classifiers considered (Section 6.1). We then focus on investigating multi-slicing and show that training on multiple time slices improves model performance (Section 6.2). We analyze how this is achieved by assessing the effects of sample size and including samples from multiple time slices (Section 6.3). Next, we discuss how the number of time slices influences predictive performance (Section 6.4).

6.1. Classifier selection

For an initial assessment of the churn prediction performance on our dataset we report 5x2-fold cross-validation results of the three benchmark classification algorithms considered. We train on a single time slice and perform out-of-period testing on the most recent time slice with an average churn rate of 7.5%. All classifiers are substantially better at predicting churn than the baseline of a random model achieving an AUC of 0.5 and a TDL of 1. Fig. 8 shows that the random forest classifier ranks first for both AUC and TDL. With random forest, churn of the B2B customers can be predicted well based on RFM features and features regarding CRM and delivery performance as the top predictors (Appendix C). The pairwise Wilcoxon signed-rank tests show that random forest performs significantly better than SVM (AUC) and logistic regression (AUC, TDL). We therefore chose this method for all following analyses.

We report out-of-period results on a more recent time slice than the one used for training to reflect the temporal nature of our setting. The impact of changing conditions on predictive performance cannot be detected when training and testing on the same time slice with out-of-sample testing.

We confirm that our setting is exposed to a non-stable environment by comparing results obtained with out-of-sample and out-of-period testing for random forest (Table 2). Out-of-period results are significantly lower, giving a more realistic estimate of classification performance and indicating that generalization to a period not yet seen is difficult.

Table 2
Random forest performance estimates (standard error) for out-of-sample and out-of-period testing methods over 5x2-fold cv.

	5x2-fold cv	
Metric	Out-of-sample	Out-of-period
AUC	0.746 (0.012)	0.704 (0.012)
TDL	3.296 (0.217)	2.856 (0.178)

Table 3
Average performance (standard error) over 5x2-fold cv for different slicing techniques; Statistically significant difference to extended single slicing: $p<0.1$(*) $p<0.05$(**) $p<0.01$(***) for Wilcoxon signed-rank test with Bonferroni correction.

Metric	Time window	Simple single-slicing	Extended single-slicing	Multi-slicing
AUC	18	0.704 (0.012)	0.704 (0.014)	0.718 (0.013) **
	21	0.704 (0.012)	0.707 (0.016)	0.725 (0.016) **
	24	0.704 (0.012)	0.704 (0.018)	0.731 (0.016) ***
	27	0.704 (0.012)	0.709 (0.018)	0.732 (0.017) ***
TDL	18	2.856 (0.178)	2.816 (0.244)	3.376 (0.259) **
	21	2.856 (0.178)	2.842 (0.204)	3.443 (0.249) ***
	24	2.856 (0.178)	2.749 (0.233)	3.697 (0.190) **
	27	2.856 (0.178)	2.829 (0.320)	3.750 (0.227) **

6.2. Performance enhancement with multi-slicing

We analyze whether multi-slicing improves predictive performance in contrast to single-slicing and assess whether extended single-slicing or multi-slicing can exploit a given time window of data better. For four different time windows of available data (18, 21, 24 and 27 months), we compare extending the feature creation window (extended single-slicing) to training on as many time slices as possible (multi-slicing). Simple single-slicing using a constant feature window of 12 months and a label window of 3 months serves as a baseline.

Table 3 shows, that training on a single slice with an extended feature window does not necessarily lead to better results than the baseline of simple single-slicing with a feature window of 12 months. Performance is on a similar level, slightly better for AUC and slightly worse for TDL. There is no significant difference between simple and extended single-slicing. Multi-slicing, however, boosts performance for all analyzed time windows. Multi-slicing results are significantly higher than extended single-slicing results for both metrics (see Table 3). The longer the time window, the more slices can be used (18 months: 4 slices, 21: 7, 24: 10 and 27: 13). Multi-slicing improves AUC by 2.1% to 4.0%, reaching values between 0.718 and 0.732. TDL is much higher with multi-slicing, exhibiting an increase between 18.2% and 31.3%. It is evident that multi-slicing exploits a given time frame better than single-slicing, leading to more accurate churn predictions.

6.3. Effects of sample size and including samples from multiple time slices

Machine learning algorithms generally perform better with more samples to train on. Therefore, the previous results lead to the following questions: What drives the performance improvement of multi-slicing? Is the improvement just due to training on a larger sample size? Or is there a unique benefit in training on observations from different time slices?

To isolate the two effects, we compare multi-slicing to a downsized version that limits the number of samples drawn from multiple time slices to that of a single slice. Downsized multi-slicing reveals the effect that stems from training on samples from different time slices. The effect of a larger training set size can be observed

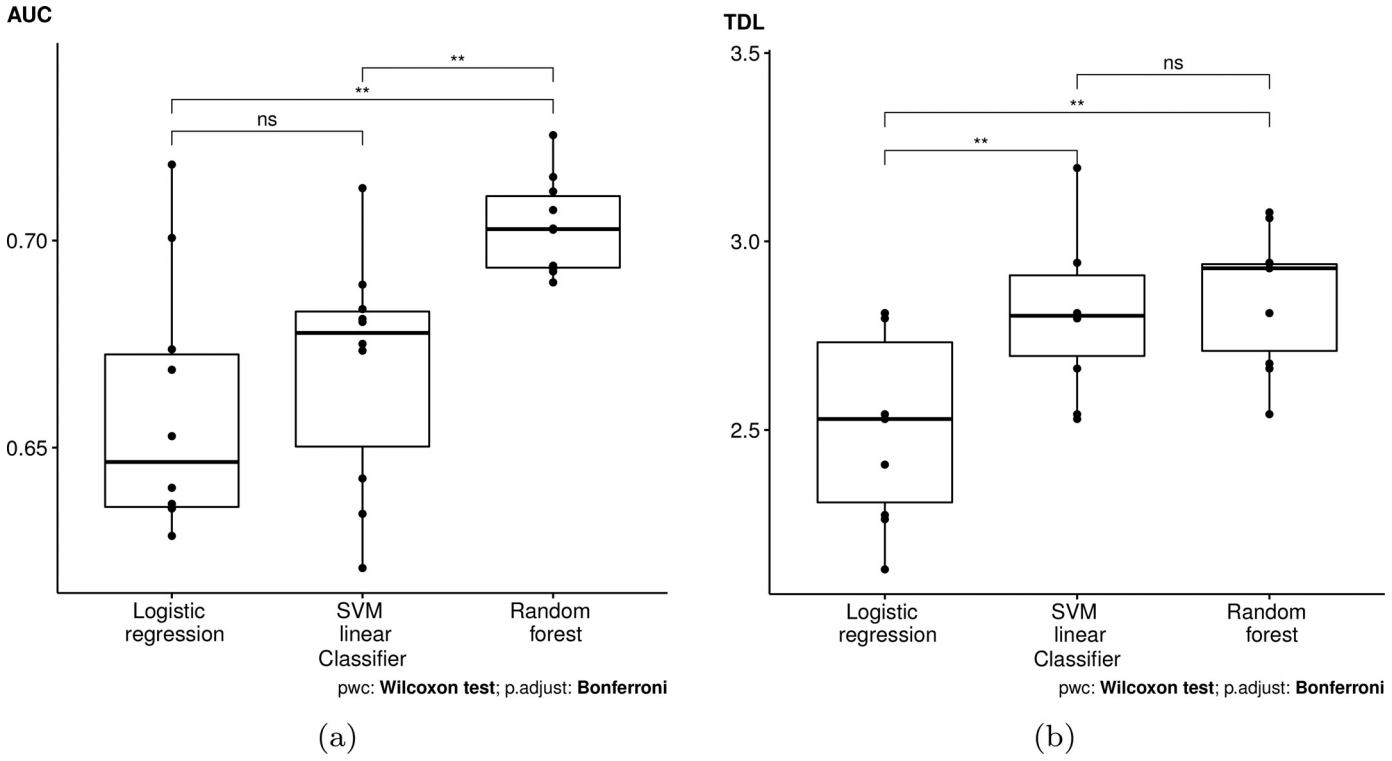

Fig. 8. Boxplots of (a) AUC and (b) TDL estimates over 5x2-fold cv; Wilcoxon signed-rank test results for pairwise comparison of classifiers ($p<0.1$(*) $p<0.05$(**) $p<0.01$(***)).

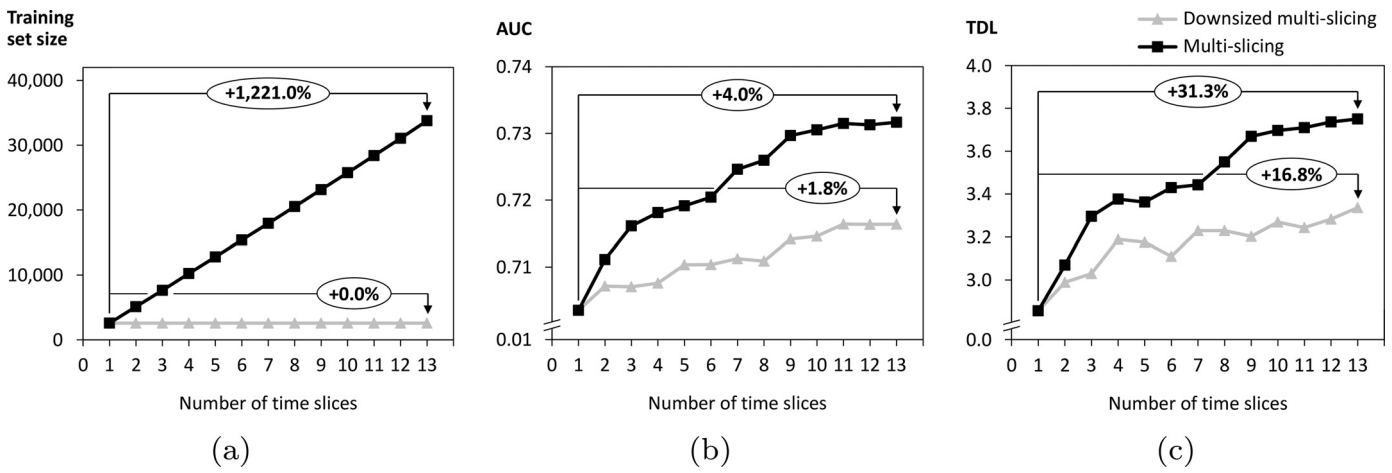

Fig. 9. Performance of AUC and TDL compared to training set size under varying number of time slices used in multi-slicing.

by evaluating the difference between the original multi-slicing and the downsized version.

Fig. 9 (a) visualizes the development in training set size for the two versions: For multi-slicing the sample size almost multiplies with the number of slices while for downsized multi-slicing it stays the same.

If the multi-slicing effect was only driven by sample size, charts (b) and (c) would show a flat line for the downsized version as in chart (a). Instead, the downsized version of multi-slicing improves AUC and TDL performance compared to single-slicing with the same sample size (represented by the point at number of time slices = 1). When testing on the latest time slice, the time span available in our dataset allows for training on a maximum of 13 time slices. Downsized multi-slicing achieves 1.8% higher AUC and 16.8% higher TDL. The original version of multi-slicing performs even better, improving AUC by 4.0% and TDL by 31.3%. We conclude that improvement by multi-slicing is driven by both: Training on samples from different time slices and increased sample size.

6.4. Impact of number of time slices

Next, we evaluate, how many time slices should be included in the training data. Looking at the development of AUC and TDL over the number of time slices in Fig. 9 (b) and (c), multi-slicing and downsized multi-slicing exhibit a similar pattern. Adding only a few recent time slices already increases performance substantially for both versions. Afterwards, not every additional slice leads to better performance but an upwards trend is observable. Performance generally increases with the number of slices, but the marginal benefit decreases. While most of the benefit can already be achieved by adding a few time slices, we recommend to include as many time slices as possible.

7. Conclusion

With a random forest churn prediction model, we compare training on multiple time slices to training on one time slice of data. The multi-slicing approach uses available data more efficiently and performs better. We compare single- and multi-slicing as well as different versions of multi-slicing to detect the underlying effects. Our results indicate that both increased sample size and training on observations from different time slices enhance predictive performance.

We show that training on multiple time slices significantly improves AUC and TDL results compared to single-slicing. A given time window of training data is better exploited by constructing multiple time slices to train on than by extending the feature time window.

We investigate two possible reasons for the performance increase: Increased sample size and training on samples from different time slices. We eliminate the sample size effect by comparing multi-slicing to a downsized version. This version still leads to a substantial performance improvement. It shows that there is unique benefit in training on observations from different time slices that goes beyond increased sample size. Multi-slicing enables training on data from more than just the most recent time slice and prevents overfitting to the situation of one specific time slice. Therefore, models can better generalize and predict future churners. The increased sample size leads to even higher improvement, addressing the absolute rarity issue by training on more churn samples from the past. While the first time slices have the strongest impact, we recommend to include as many time slices as possible.

While training and testing on only one time slice is still the most common approach in churn prediction we highly recommend to test a churn prediction model beyond the training time frame. This assesses performance more reliably in situations with a changing environment. Under such conditions, our results demonstrate that multi-slicing is particularly valuable and obtains good out-of-period performance results.

Going forward, it would be interesting to analyze whether our results generalize to other churn prediction problems and even to classification problems from other domains relying on time series data. When the problem situation varies over time, training on multiple time slices makes models more robust. Multi-slicing achieves better predictive performance by training on more samples and increasing the generalization capability. However, further studies need to be conducted to verify the effect on different case data.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ejor.2021.05.035

References

Anderson, E., & Weitz, B. (1989). Determinants of continuity in conventional industrial channel dyads. *Marketing Science, 8*(4), 310–323.

Ballings, M., & Van den Poel, D. (2012). Customer event history for churn prediction: How long is long enough? *Expert Systems with Applications, 39*(18), 13517–13522.

Ballings, M., & Van Den Poel, D. (2015). CRM in social media: Predicting increases in Facebook usage frequency. *European Journal of Operational Research, 244*(1), 248–260.

Bose, I., & Chen, X. (2009). Quantitative models for direct marketing: A review from systems perspective. *European Journal of Operational Research, 195*(1), 1–16.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Buckinx, W., & Van den Poel, D. (2005). Customer base analysis: Partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting. *European Journal of Operational Research, 164*(1), 252–268.

Burez, J., & Van den Poel, D. (2007). CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications, 32*(2), 277–288.

Burez, J., & Van den Poel, D. (2008). Separating financial from commercial customer churn: A modeling step towards resolving the conflict between the sales and credit department. *Expert Systems with Applications, 35*(1-2), 497–514.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36*(3), 4626–4636.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chen, C., Liaw, A., & Brieman, L. (2004). Using random forest to learn imbalanced data: Technical Report No. 666. University of California, Berkley. *Using Random Forest to Learn Imbalanced Data, 110*(1–12), 12.

Chen, K., Hu, Y.-H., & Hsieh, Y.-C. (2015). Predicting customer churn from valuable B2B customers in the logistics industry: A case study. *Information Systems and e-Business Management, 13*(3), 475–494.

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research, 223*(2), 461–472.

Coussement, K., & De Bock, K. W. (2013). Customer churn prediction in the online gambling industry: The beneficial effect of ensemble learning. *Journal of Business Research, 66*(9), 1629–1636.

Coussement, K., Lessmann, S., & Verstraeten, G. (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. *Decision Support Systems, 95*, 27–36.

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications, 34*(1), 313–327.

Coussement, K., & Van den Poel, D. (2009). Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Systems with Applications, 36*(3), 6127–6134.

De Bock, K. W., & Van Den Poel, D. (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications, 39*(8), 6816–6826.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research, 269*(2), 760–772.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923.

Egan, J. P. (1975). Signal detection theory and ROC-analysis. In Series in cognition and perception. Academic press.

Farquad, M. A., Ravi, V., & Raju, S. B. (2014). Churn prediction using comprehensible support vector machine: An analytical CRM application. *Applied Soft Computing Journal, 19*, 31–40.

García, D. L., Nebot, À., & Vellido, A. (2017). Intelligent data analysis approaches to churn as a business problem: A survey. *Knowledge and information systems, 51*(3), 719–774.

Glady, N., Baesens, B., & Croux, C. (2009). Modeling churn using customer lifetime value. *European Journal of Operational Research, 197*(1), 402–411.

Gür Ali, Ö., & Arıtürk, U. (2014). Dynamic churn prediction framework with more effective use of rare event data: The case of private banking. *Expert Systems with Applications, 41*(17), 7889–7903.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1–3), 389–422.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis, 6*(5), 429–449.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.

Kumar, D. A., & Ravi, V. (2008). Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies, 1*(1), 4–28.

Larivière, B., & Van Den Poel, D. (2004). Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services. *Expert Systems with Applications, 27*(2), 277–285.

Larivière, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications, 29*(2), 472–484.

Lessmann, S., & Voß, S. (2009). A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research, 199*(2), 520–530.

Leung, H. C., & Chung, W. (2020). A Dynamic Classification Approach to Churn Prediction in Banking Industry. In Amcis 2020 proceedings data science and analytics for decision support (sigdsa). Association for Information Systems.

Miguéis, V., Camanho, A., & Falcão e Cunha, J. (2013). Customer attrition in retailing: An application of Multivariate Adaptive Regression Splines. *Expert Systems with Applications, 40*(16), 6225–6232.

Miguéis, V., Van den Poel, D., Camanho, A., & Falcão e Cunha, J. (2012). Modeling partial customer churn: On the value of first product-category purchase sequences. *Expert Systems with Applications, 39*(12), 11250–11256.

Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., & Mason, C. H. (2006). Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research, 43*(2), 204–211.

Óskarsdóttir, M., Bravo, C., Verbeke, W., Sarraute, C., Baesens, B., & Vanthienen, J. (2017). Social network analytics for churn prediction in telco: Model

building, evaluation and network architecture. *Expert Systems with Applications, 85*, 204–220.
Piatetsky-Shapiro, G., & Masand, B. (1999). Estimating campaign benefits and modeling lift. In *Proceedings of the fifth ACM SIGKDDinternational conference on knowledge discovery and data mining - KDD '99* (pp. 185–193). New York, New York, USA: ACM Press.
Reichheld, F. F. (1996). Learning from customer defections. *Harvard business review, 74*(2), 56–69.
Risselada, H., Verhoef, P. C., & Bijmolt, T. H. (2010). Staying power of churn prediction models. *Journal of Interactive Marketing, 24*(3), 198–208.
Seppälä, T., & Thuy, L. (2018). A combination of multi-period training data and ensemble methods to improve churn classification of housing loan customers. In *Proceedings of the 2nd international conference on advanced research methods and analytics (CARMA 2018)* (pp. 141–144). Universidad Politècnica de València.
Somol, P., Baesens, B., Pudil, P., & Vanthienen, J. (2005). Filter- versus wrapper-based feature selection for credit scoring. *International Journal of Intelligent Systems, 20*(10), 985–999.
Tamaddoni Jahromi, A., Stakhovych, S., & Ewing, M. (2014). Managing B2B customer churn, retention and profitability. *Industrial Marketing Management, 43*(7), 1258–1268.
Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, NY: Springer.
Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research, 218*(1), 211–229.
Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications, 38*(3), 2354–2364.
Wei, C.-P., & Chiu, I.-T. (2002). Turning telecommunications call details to churn prediction: A data mining approach. *Expert Systems with Applications, 23*(2), 103–112.
Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations, 6*(1), 7–19.
Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin 1, 6*, 80–83.
Zahavi, J., & Levin, N. (1997). Applying neural computing to target marketing. *Journal of Direct Marketing, 11*(1), 5–22.
Zeithaml, V. A., Berry, L. L., & Parasuraman, A. (1996). The behavioral consequences of service quality. *Source: Journal of Marketing, 60*(2), 31–46.
Zhu, B., Baesens, B., Backiel, A., & Vanden Broucke, S. K. (2017). Benchmarking sampling techniques for imbalance learning in churn prediction. *Journal of the Operational Research Society, 69*(1), 49–65.