European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

Predictive analytics for customer repurchase: Interdisciplinary integration of buy till you die modeling and machine learning

Ping Chou[a], Howard Hao-Chun Chuang[a,*], Yen-Chun Chou[a], Ting-Peng Liang[b]

[a] Department of Management Information Systems, National Chengchi University, Taipei, Taiwan
[b] Electronic Commerce Research Center, National Sun Yat-Sen University, Kaohsiung, Taiwan

ARTICLE INFO

Article history:
Received 12 February 2020
Accepted 9 April 2021
Available online xxx

Keywords:
Analytics
Customer repurchase
'Buy till You Die'
Lasso
Machine Learning

ABSTRACT

Predicting customer repurchase propensity/frequency has received broad research interests from marketing, operations research, statistics, and computer science. In the field of marketing, Buy till You Die (BTYD) models are perhaps the most representative techniques for customer repurchase prediction. Those probabilistic models are parsimonious and typically involve only recency and frequency of customer activities. Contrary to BTYD models, a distinctly different class of predictive models for customer repurchase is machine learning. This class of models include a wide variety of computational and statistical learning algorithms. Unlike BTYD models built on low-dimensional inputs and behavioral assumptions, machine learning is more data-driven and excels at fitting predictive models to a large array of features from customer transactions. Using a large online retailing data, we empirically assess the prediction performance of BTYD modeling and machine learning. More importantly, we investigate how the two approaches can complement each other for repurchase prediction. We use the BG/BB model given the discrete and non-contractual problem setting and incorporate BG/BB estimates into high-dimensional Lasso regression. In addition to showing significant improvement over BG/BB and Lasso without BG/BB, the integrated Lasso-BG/BB provides interpretability and identifies BG/BB predictions as the most influential feature among ~100 predictors. The lately developed CART-artificial neural networks exhibit similar patterns. Robustness checks further show the proposed Lasso-BG/BB outperforms two sophisticated recurrent neural networks, validating the complementarity of machine learning and BTYD modeling. We conclude by articulating how our interdisciplinary integration of the two modeling paradigms contributes to the theory and practice of predictive analytics.

1. Introduction

In highly competitive consumer markets, predicting customer repurchase propensity/frequency has received broad research interests from marketing, operations research, statistics, and computer science (De Caigny, Coussement, & De Bock, 2018; Platzer & Reutterer, 2016). Repurchase prediction has been a focal issue in personalized marketing as well as customer base analytics for many years (Chou & Chuang, 2018; Martínez, Schmuck, Pereverzyev Jr, Pirker, & Haltmeier, 2020; Suh, Lim, Hwang, & Kim, 2004). Identifying who are likely to purchase within the next week/month/quarter from a large customer base is a penial problem in data analytics because it allows managers to allocate sales resources and launch marketing campaigns more efficiently. On the business side, this issue has become increasingly important with the advent of multiple types of online platforms, all of which have to understand customer behaviors and predict customer activities based on transactional data. On the methodological front, machine learning has become increasingly popular for repurchase prediction recently (Martínez et al., 2020). Nonetheless, there is still continual development of marketing models for this prediction task (Dew & Ansari, 2018; Gopalakrishnan, Bradlow, & Fader, 2017).

In the marketing literature, the Buy till You Die (BTYD) models are perhaps the most representative techniques for customer repurchase prediction. Based on the transaction *setting* – contractual versus non-contractual – and transaction *timing* – continuous versus discrete, marketing researchers have developed an array of *probability models* for each of the four *setting-timing* combinations (Fader & Hardie, 2009). Various probability distributions are used to characterize customer lifetime, purchase intensity, and so forth to model a repeated purchase and dropout processes. Those probabilistic models are parsimonious and typically involve only

* Corresponding author.
E-mail addresses: chuang@nccu.edu.tw (H.H.-C. Chuang), yenchun@nccu.edu.tw (Y.-C. Chou), tpliang@mail.nsysu.edu.tw (T.-P. Liang).

https://doi.org/10.1016/j.ejor.2021.04.021
0377-2217/

recency and frequency of customer activities. Well-known BTYD models include Pareto/NBD (Schmittlein, Morrison, & Colombo, 1987) and BG/NBD (Fader, Hardie, & Lee, 2005) for non-contractual continuous timing, and BG/BB (Fader, Hardie, & Shang, 2010) – for non-contractual discrete timing, etc. The non-contractual setting is particularly challenging because it is difficult to closely observe customer status, and models have to be built upon transaction records. Note that the widely seen customer churn prediction is tightly coupled with repurchase prediction in non-contractual settings (Buckinx & Van den Poel, 2005; Miguéis, Van den Poel, Camanho, & e Cunha, 2012), where churn is a latent status and has to be inferred from repurchase incidences or BTYD estimate.[1] While some use these two terms interchangeably in the literature, we use repurchase prediction that more precisely matches our research objective.

Contrary to the BTYD models that attempt to explicitly model behavioral processes through probability distributions, a distinctly different class of models for customer prediction tasks is *machine learning*. Note that machine learning here covers not only *statistical machine learning*, but also supervised/unsupervised learning algorithms from computer science. Unlike BTYD models that rely on few inputs (i.e., recency and frequency as sufficient statistics) and behavioral assumptions, machine learning takes a data-driven approach to predictive modeling. While machine learning can uncover patterns from low-dimensional inputs too, this modeling approach is distinctly powerful in being able to extract information from high-dimensional customer features (Martínez et al., 2020). Learning algorithms such as linear/logistic regression, decision trees, random forests, support vector machines, and artificial neural networks (ANN) are then employed to generate predictions (De Caigny et al., 2018; Verbeke, Martens, Mues, & Baesens, 2011).

Unlike BTYD models with parameter estimates that represent customer heterogeneity, purchase intensity, and churn propensity, popular machine learning models (e.g., ANN, support vector machines) typically have no such transparency. When it comes to predictive analytics for customer repurchase propensity/frequency, despite that both avenues have the same objective, researchers and practitioners still have limited understanding about whether the two classes of modeling approaches can cross-fertilize each other. Would BTYD models built on simple statistics and behavioral assumptions be complementary to algorithms learnt from various features? Alternatively, do the BTYD models simplify customer purchase and dropout processes too much and ignore nuanced information hidden in other features to be explored by machine learning algorithms? Moreover, how can the BTYD models' estimates be integrated into machine learning approaches? We are intrigued by the foregoing issues that are relevant to many industry sectors and yet fully addressed. Hence, we conduct an empirical study in which we compare and assess the complementarity of the two avenues.

For our empirical investigation, we collect a large data set that has transaction records of an anonymous online retailer with over 500,000 members and over 1.25 million transaction records over 33 months. The research goal is to develop data-driven approaches that help the online retailer predict whether each of its members will return and purchase in the next quarter. To address the practical need and research questions discussed above, for the BTYD modeling approach, we calibrate a BG/BB (Beta-Geometric/Beta-Bernoulli) model (Fader et al., 2010) in line with our non-contractual data and discrete problem setting. The BG/BB model has been widely applied and proven effective for such repurchase/churn prediction tasks (McCarthy, Fader, & Hardie, 2016; Zhang, 2008). For the machine learning approaches, we extract around 100 features (e.g., recency, frequency, monetary (RFM), return/cancel, tool of payment, and device) from customer transaction records pertaining to multiple dimensions as predictor variables. To assess whether the two approaches can complement each other, we incorporate prediction values and behavior estimates of the BG/BB model into learning processes. Specifically, we adopt the Lasso regression method originated from statistical machine learning (Tibshirani, 1996). Lasso modeling has appeared in prior machine learning studies (e.g., Cui, Rajagopalan & Ward, 2020, Martínez et al., 2020) and has become an impactful technique in predictive modeling. Instead of going directly for other powerful learning algorithms with lower transparency, we begin with Lasso not only because of its ability to shrink effect sizes of non-effective predictors, but also due to its theoretical robustness to member-quarter panel data that could violate i.i.d. assumptions imposed by numerous learning algorithms (Medeiros & Mendes, 2016). Also, owing to the transparency of Lasso regression, it would be intriguing to see how the BG/BB estimates are ranked among other features directly calculated from data.

To our surprise, despite having the luxury of exploiting information carried by high-dimensional input features, Lasso regression, without BG/BB estimates, does not outperform the BG/BB model with merely recency and frequency as inputs. Nonetheless, incorporating BG/BB estimates into Lasso regression leads to a significant improvement in prediction performance. Moreover, the variables with non-negligible effects identified by Lasso are almost exclusively BG/BB outputs and related to old-fashioned RFM. Intrigued by the performance enhanced by adding BG/BB estimates into high-dimensional Lasso regression, we further apply feedforward ANN with more flexible/complicated functional forms. By feeding BG/BB into ANN, we find that the integrated model consistently leads to the best performance. Further, considering time dependencies of panel data, we conduct robustness checks and show that the proposed Lasso-BG/BB outperforms two sophisticated recurrent neural networks.

Our modeling effort makes major contributions to the literature and practice. First, based on an empirical study on longitudinal online customer repurchase behaviors, we show that machine learning models exposed to high-dimensional features do not necessarily outperform a low-dimensional BTYD model when it comes to predicting customer repurchase. Moreover, parameter estimates and prediction outputs of the BTYD model emerge as the most influential predictors of the proposed Lasso-BG/BB, supporting that BTYD would not be entirely overtaken by machine learning approaches and could enhance their performance. By offering empirical evidence on the complementarity of BTYD modeling and machine learning, we present a conversation-provoking study to stimulate more integrative work of the two schools of models. Second, we show that the parsimonious Lasso regression, with BG/BB estimates as input features, performs well compared to more complicated feedforward and recurrent neural networks without full transparency. The research findings are highly relevant to numerous practitioners who prefer to adopt simple, stable, and interpretable prediction models for consumer marketing analytics. By leveraging the BG/BB model and readily available data, the proposed Lasso-BG/BB is easy-to-implement, with much lower computing costs than ensemble/deep learning models that might result in marginal improvement for prediction tasks of non-perpetual data (Rudin & Carlson, 2019).

2. Literature review

Application of probability models, BTYD models in particular, to customer repurchase/churn has been investigated by both marketing and operations researchers for decades, and most of those models treat observed transactions as the outcome of un-

[1] P(Active)/P(Alive) estimates from BTYD models indicate repurchase/churn likelihood. Alternatively, when repurchase events of a customer remain zero for a period of time, he/she could be labelled as "churn".

derlying stochastic processes (Fader & Hardie, 2009; Gupta et al., 2006). Note that we cover studies on customer repurchase as well as churn prediction because the two terms are bundled in non-contractual settings (see Introduction). In the marketing literature, probability models for customer repurchase are classified into four categories based on the transaction timing – continuous versus discrete – and setting – contractual versus non-contractual (Fader & Hardie, 2009). Those models are especially well-known for their non-contractual applications, where the uncertainty is higher than the contractual setting due to the lack of observational evidence of churn (Reinartz & Kumar, 2000). BTYD is arguably the most representative class of low-dimensional probability models for customer repurchase prediction. A BTYD model is formulated by a probabilistic mixture conditional on a set of distributional assumptions regarding customer lifetime and purchase intensity. Specifically, the class of BTYD models share the common idea of modeling the transaction process, dropout process, and consumer heterogeneity with the aid of various probability distributions.

Among the BTYD models, the Pareto/NBD (Schmittlein et al., 1987) for continuous, non-contractual data is the seminal and perhaps the most famous BTYD model. The BG/NBD is a well-known alternative to Pareto/NBD due to its lower computational difficulties (Fader et al., 2005). For discrete, non-contractual data, the BG/BB (Fader et al., 2010) is a representative probability model. The afore-mentioned models have motivated a lasting stream of subsequent studies that either introduce alternative distributions (e.g., Batislam, Denizel, & Filiztekin, 2007, Jerath, Fader, & Hardie, 2011) or add a dimension on top of recency and frequency (e.g., Gopalakrishnan et al., 2017, Platzer & Reutterer, 2016, Reutterer, Platzer, & Schröder, 2020, Zhang, Bradlow, & Small, 2015). In spite of the seemingly complicated mixture distributions for customer visits/transactions over discrete or continuous time periods, BTYD models are simple in that they require only two sufficient statistics – purchase recency and frequency – as data inputs for modeling customer response. Despite that few BTYD models (e.g., Abe, 2009, Schweidel & Knox, 2013) attempt to quantify effects of covariates such as initial purchase amount on estimated transaction/dropout rates, advances in BTYD models primarily come from employing generic probability distributions to customer heterogeneities. Accordingly, studies that aim for improving well-known BTYD models are usually benchmarked against probability models only (e.g., Batislam et al., 2007, Gopalakrishnan et al., 2017, Jerath et al., 2011, Platzer & Reutterer, 2016, Shi, Chen, & Sethi, 2019, Van Oest & Knox, 2011), leaving research opportunities for empirically comparing/integrating BTYD models and machine learning models that consider much higher input dimensions.

In contrary to BTYD models with explicit behavioral and probabilistic assumptions on transaction, dropout, and heterogeneity, a majority of machine learning models for customer repurchase/churn in the literature rely on uncovering patterns from high-dimensional features using statistical learning and computational algorithms (Martínez et al., 2020). The popular machine learning models (e.g., ANN, random forest, support vector machines) used to generate predictions are almost 'black box'. Recently, more sophisticated techniques such as gradient boosting machines (Lemmens & Gupta, 2020; Martínez et al., 2020; Milošević, Živić, & Andjelković, 2017) and recurrent neural networks (Alboukaey, Joukhadar, & Ghneim, 2020; Mena, De Caigny, Coussement, De Bock, & Lessmann, 2019; Wang, Lai, Zhang, Wang, & Chen, 2020) have been employed to predict customer repurchase/churn. The predictive power of those algorithms also comes at a price of limited interpretability. However, methods with higher interpretability, e.g., classification tree and logistic regression, are usually outperformed by near black box models (De Caigny et al., 2018). The advantage of brute-force learning algorithms lies in their capability of extracting hidden patterns from data with fewer assumptions about underlying customer transaction processes (Chatfield, 1995). However, this modeling approach comes at costs of computational complexities and more noise embedded in correlated/irrelevant inputs (Hadden, Tiwari, Roy, & Ruta, 2007; Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012). In terms of predicting customer repurchase/churn, machine learning is increasingly popular for predictive analytics in marketing and operations. For brevity, we refer readers to studies with extensive literature review (De Caigny et al., 2018; Gupta et al., 2006; Verbeke et al., 2011, 2012) on the application of machine learning in repurchase/churn prediction.

From reviewing related work, we find that the two streams of modeling approaches, in spite of their shared goal of predicting customer behavior, are fairly isolated in the literature. The hard core marketing modelers attempt to push the performance boundary of BTYD models by tweaking probabilistic assumptions (e.g., Reutterer et al., 2020, Shi et al., 2019), whereas many other researchers (Baesens, Viaene, Van den Poel, Vanthienen, & Dedene, 2002; De Caigny et al., 2018; Keramati et al., 2014; Verbeke et al., 2012) broadly expand inputs and use complex learning algorithms for the sake of boosting prediction accuracy. Few studies have compared the performance of rudimentary learning algorithms – regression model (Hopmann & Thede, 2005; Zhang, 2008) and decision tree (Jahromi, Stakhovych, & Ewing, 2016) – to BTYD. To the best of our knowledge, Tamaddoni, Stakhovych, and Ewing (2016) is the first to compare the predictive performance of sophisticated machine learning (i.e., support vector machines and boosting) to BTYD modeling. They find machine learning outperforms Pareto/NBD under most of the tested circumstances (varying sample size, purchase frequencies, and churn ratio). However, their comparisons are performed in a low-dimensional setting with only frequency, recency, and total time observed as input features. More recently, some studies compare BTYD models with ANN (Chen, Guitart, del Río, & Periánez, 2018; Xie & Huang, 2020). While Chen et al. (2018) find that ANN based on high-dimensional customer behavior variables outperform BTYD based on RFM variables in predicting customer lifetime value, Xie and Huang (2020) find that ANN and BTYD perform almost identically when only the above-mentioned low-dimensional variables are available. In summary, machine learning using low-dimensional inputs empirically tends to perform at least as good as BTYD, and usually outperforms BTYD when high-dimensional information is available. Nonetheless, all of the foregoing studies assess the performance of BTYD models for continuous problem setting except Zhang (2008), leaving research opportunities to further contrast the two modeling streams under discrete timing (e.g., BG/BB). Besides, instead of using learning algorithms typically developed for cross-sectional data, we deliberately employ Lasso regression that is robust to panel data, which enables researchers to better capture the temporal evolution of customer behaviors for repurchase/churn prediction (Holtrop, Wieringa, Gijsenberg, & Verhoef, 2017).

In addition to model comparisons, a few studies (Schwartz, Bradlow, & Fader, 2014; Xie, 2019) apply machine learning algorithms to facilitate BTYD model selection and parameter estimation. Specifically, Schwartz et al. (2014) propose a tree-based model, which takes summary statistics as inputs, to select a type of BTYD models that best fit a given transaction dataset. The model is highly interpretable and does not require one to fit multiple candidate models to data beforehand. Xie (2019) uses ANN that take time-varying recency and frequency as inputs to predict BTYD model (Pareto/NBD) parameters, and uses the estimates to predict purchase frequency from the expectation expression of Pareto/NBD, achieving lower prediction errors than BTYD. Prior studies on integrating the two modeling streams mainly apply machine learning to model selection and updating, and make predictions based on BTYD formulations. The parametric assumptions make BTYD par-

simonious in its design, and exhibit high transparency and interpretability (De Caigny et al., 2018; Gupta et al., 2006). However, BTYD modeling is usually limited to sufficient statistics and does not fully utilize input features beyond RFM (Abe, 2009; Schweidel & Knox, 2013).

Considering the advance of collecting data in nowadays, how to integrate BTYD models with high-dimensional data on behaviors related to cancel, return, delivery, payment, and discount is under-explored. A modeling approach that integrates BTYD and machine learning would enable one to better leverage information in collected data and lead to a more flexible prediction engine. Hence, we complement the literature by proposing a novel combination of BG/BB and Lasso regression, a powerful statistical machine learning algorithm for high-dimensional data, under the context of online retailing. We choose the powerful yet easy-to-implement Lasso regression because it allows us to maintain high model transparency and it can be scaled up to a large number of input features, a critical attribute that BTYD models are short of. In addition to showing the efficacy of this hybrid Lasso-BG/BB modeling approach, our proposition is triangulated via combining BG/BB and ANN. Grounded in a real-world case, our analysis offers insights that have not been reported in the literature and aims to prompt more interactions between the two predictive modeling paradigms.

3. Data and features

We use the dataset provided by one of the biggest online retailing service providers in Taiwan. The service provider offers online storefronts and associated IT infrastructures for merchants to open online stores simply by a few clicks. This setting serves our study well because the service provider has been actively trying to develop prediction models for customer repurchase as a value-added service to merchants. The company considers BTYD and machine learning as contenders for their prediction tasks. We obtain the representative dataset of an anonymous merchant (the retailer hereafter) who utilizes the online retailing service to sell clothing and accessories for almost three years (33 months from May 2015 to January 2018). The dataset is composed of 1,250,587 transaction records and a total of 532,410 members. Each transaction may contain more than one product and each record is composed of member id, cart id, and product id. The cart id represents a transaction where goods are bought together in a basket.

Accordingly, we first compute the transaction-level summary statistics by aggregating transaction records with the same member id and cart id. In addition to covering purchase-related RFM, the transaction records encompass cancel events, return events, delivery choices, payment choices, and discount usage. Via the interview with managers of the service provider, they note that the retailer usually plans marketing campaigns by quarters due to the nature of the apparel goods. As a result, our goal is to help the retailer predict whether a member would repurchase in the next quarter. To meet the operational requirements in our research site, we collapse transaction-level details of each member into features that reflect his/her state across different quarters (to be introduced below). Starting from a member's first active quarter, we consistently compute and fill in his/her state metrics for all subsequent quarters and detect no missing value in the afore-mentioned activity dimensions. Details of raw data pre-processing can be found in the Appendix A.

Fig. 1 shows the number of total members (left panel) and new members (right panel) of the retailer over eleven quarters (a quarter is composed of three months). We can observe that the number of new members increases from quarter 1 to quarter 6, especially during the period from quarter 4 to quarter 6. Since then, the number of new members gradually decreases over quarters but the total number of members still increases. Other than members, we look into the summary of transactions over eleven quarters (see Table 1). The number of transactions grows and peaks at the quarter 5, and starts to decline since then. Interestingly, while the number of orders decreases in the later periods, the average number of products bought and amounts spent per order increases over time. We also find 55% of members in our dataset are one-time buyers in the observation periods. The pattern seems common in online stores as alternative options are just a few click away. The service provider is concerned that many members of the retailer are probably short-lived. Thus, it is critical to the retailer to assess customer response and find out who are the returning customers.

For each quarter t, we characterize a customer's purchase behaviors in the current quarter (t) using 34 quarter-varying features (see Table 2), which go beyond purchase-related RFM and cover multiple aspects of transaction activities. Among the features, 28 (including RFM) are directly created from the aggregation of his/her transactional records in the focal quarter. These features have covered many aspects of customer purchasing behavior, i.e., transaction, device, payment, promotion, pickup, order return, and cancelation. In addition to ordinary RFM-related features, we refer to Martínez et al. (2020) and create six extra features related recency and frequency. They are *mean time between purchases* (*MTBP*), *standard deviation of time between purchases* (*STBP*) and *buying pattern* (*BP*) that encompasses 4 binary indicators. These features are effective indicators of purchase regularities/potential churns, and are defined as below:

$$MTBP_i \triangleq \sum_{k=2}^{\tau_i} \frac{\Delta t_{k,i}}{\tau_i - 1};\ STBP_i \triangleq \sqrt{\sum_{k=2}^{\tau_i} \frac{\left(\Delta t_{k,i} - MTBP_i\right)^2}{\tau_i - 1}}$$

$$BP_i \triangleq \begin{cases} normal,\ if\ (Recency_i) \le Thres_{i,1} \\ attrition,\ if\ Thres_{i,1} < (Recency_i) \le Thres_{i,2} \\ at\ risk,\ if\ Thres_{i,2} < (Recency_i) \le Thres_{i,3} \\ churn,\ otherwise \end{cases}$$

where τ stands for the number of active quarters (i.e., a quarter with at least one transaction) up to the current quarter, so the maximum value of τ is eleven. The subscript k and i respectively denote the id of active quarter and the id of customer. For any member in a given quarter, we use $\Delta t_{k,i}$ to denote the time interval between the k-th and the (k-1)-th active quarter of the ith customer, and calculate the mean and standard deviation of $\Delta t_{k,i}$ up to the quarter as $MTBP_k$ and $STBP_k$. Note that the index k starts from two because we have no information before the first transaction of any member. We define $Thres_{i,\delta} : MTBP_i + \delta \cdot STBP_i$ as the critical value, and use $Thres_{i,\delta}$ to classify the time interval since the last purchase (i.e., the so-called *Recency* in the literature) into four types (i.e., normal, attrition, at-risk, and churn) denoted by BP_i composed of four binary variables. As for one-time buyers, we set their $MTBP_i$ and $STBP_i$ to -1, and set all BP_i variables to zeros to indicate the features are not accessible.

Fig. 2 shows the panel structure of n customers by T quarters, composed of all features and the target. *Quarterly Data* refers to the 34 features in Table 2 which only account for the total amount in the current quarter t without considering differences in the number of transactions being made, e.g., a customer spending $500 in four transactions is different from another spending $2000 in one transaction. A well-known measure to eliminate such ambiguity is the *Monetary* from RFM analysis (i.e., the average payment per transaction), which highlights the importance of average statistics. So we create 26 features (x_{35}-x_{60}) of *Transaction Average* from dividing quarterly data (x_3-x_{28}) by the number of transactions (x_2). Moreover, since the above-mentioned features only reflect a member's behaviors in quarter t, it is natural to consider past

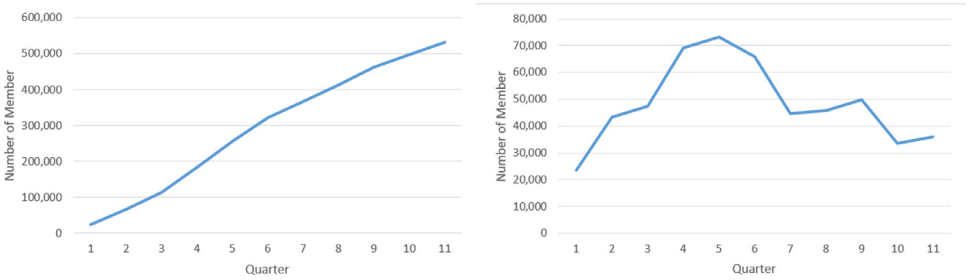

Fig. 1. Total (left) and new (right) members over eleven quarters.

Table 1
Summary of transactions over quarters.

Quarter	1	2	3	4	5	6
Number of Transactions	29,217	65,513	86,611	129,802	159,362	152,973
Avg. Number of Products	3.94	3.67	3.58	3.87	4.05	4.02
Avg. Amounts($)	766.64	845.69	880.61	887.76	883.48	949.76
Quarter	7	8	9	10	11	
Number of Transactions	122,308	132,804	146,814	112,205	112,979	
Avg. Number of Products	4.24	4.24	4.53	4.11	4.55	
Avg. Amounts ($)	1039.87	963.54	1014.59	1016.54	1133.53	

Table 2
Overview of derived features.

	Description		Description
x_1	Months since last transaction (Recency)[a]	x_{18}	Number of transactions with canceled items
x_2	Number of transactions (Frequency)	x_{19}	Number of cart-product pairs canceled
x_3	Number of transactions in holiday	x_{20}	Total cancellation-induced refund (in NT$)
x_4	Number of unique cart-product pairs	x_{21}	Number of transactions with returned items
x_5	Number of purchased units	x_{22}	Number of cart-product pairs returned
x_6	Total payment amount (Gross Monetary)[b]	x_{23}	Total return-induced refund (in NT$)
x_7	Number of transactions paid before pickup	x_{24}	Total net payment amount (Net Monetary)
x_8	Number of transactions paid by ATM	x_{25}	Number of retailer-caused cancel/return
x_9	Number of transactions paid by credit card	x_{26}	Number of customer-caused cancel/return
x_{10}	Number of transactions paid in stores	x_{27}	Number of delivery-caused cancel/return
x_{11}	Number of pickups in convenience stores	x_{28}	Number of other-caused cancel/return
x_{12}	Number of cart-product pairs with discount	x_{29}	Mean time between purchases (MTBP)
x_{13}	Total discount amount (in NT$)	x_{30}	Standard deviation of time between purchases (STBP)
x_{14}	Number of cart-product pairs with coupon	x_{31}	Indicator for recency type (BP: normal)
x_{15}	Total coupon discount (in NT$)	x_{32}	Indicator for recency type (BP: attrition)
x_{16}	Number of gifts received	x_{33}	Indicator for recency type (BP: at-risk)
x_{17}	Number of transactions via PC	x_{34}	Indicator for recency type (BP: churn)

[a] Recency is the time interval (in months) between last purchase and the ending month of current quarter. For instance, if a customer makes just one purchase in the first month of the first quarter, his/her recency in the tenth quarter will be 29 months.

[b] We are aware that the monetary measure in RFM is defined as the average spending per transaction, and do calculate a series of features related to transaction average (x_{35}-x_{60}), including the monetary measure.

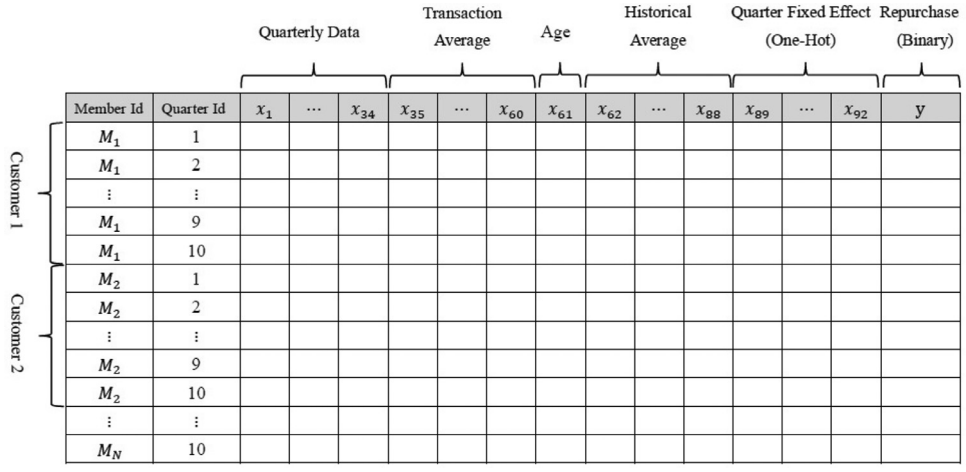

Fig. 2. The panel data structure of features and target for machine learning.

transaction records of a customer. That is, a customer can be evaluated by his/her accumulated consumptions too. Thus, we calculate a customer's *Age* (x_{61}) in terms of the number of quarters since his/her first purchase. We then obtain a customer's cumulative average value (*Historical Average*) from dividing cumulative value of x_2-x_{28} up to the measured quarter by *Age* for each member (x_{62}-x_{88}). We further include one-hot encoding for quarters (x_{89}-x_{92}) to control for time fixed effects (*Quarter*). Finally, *Repurchase* is the binary prediction target indicating whether a customer would repurchase next quarter ($t+1$) or not. Overall, we use the unbalanced panel data structure (not all members have $T=10$ records) for machine learning algorithms to approximate functional relationships between *Repurchase* and 92 features. More, despite having more than 2.2 million observations in our training set, to alleviate potential influences of outliers, we apply Winsorization to transform all input features in line with De Caigny et al. (2018).

4. Base models, evaluation metrics, and experimental settings

4.1. Lasso and BG/BB

From the feature engineering, each customer is represented by 92 features indicating his/her behaviors in the current quarter and accumulations over the quarters (see Fig. 2). With these features, we then apply the high-dimensional machine learning to predict customer repurchase in the next quarter. Instead of directly going for black-box learning algorithms, we begin with interpretable regression modeling for the sake of better interpretability. Specifically, we apply the logistic regression with Lasso regularization (Tibshirani, 1996) to force coefficients of non-effective predictors be zero, such that we can identify influential predictors clearly. Being able to handle high-dimensional inputs and alleviate overfitting, Lasso regularization in linear and non-linear regression has become an impactful technique and a prosperous research area. Unlike many supervised learning algorithms are developed under i.i.d. data assumptions, Lasso regression is theoretically valid for longitudinal data with time dependencies. Medeiros and Mendes (2016) prove that the Adaptive Lasso, which generalizes ordinary Lasso, exhibits model selection consistently in high-dimensional longitudinal data. The consistency holds even when error terms are non-Gaussian and heteroskedastic. Smeekes and Wijler (2018) further provide evidence in simulation/empirical data that Lasso regression models outperform other standard econometric models for predicting non-stationary macroeconomic time-series. Hence, we posit that the Lasso logit regression is robust for our prediction tasks using member-quarter panel data, while satisfying our goal of directly uncovering feature importance. The following is the logit log-loss function for binary Lasso regression:

$$-\sum_{i=1}^{n}\left[y_i\cdot\log\left(\frac{1}{1+e^{-\left(\beta_0+\sum_{j=1}^{p}\beta_j x_{ij}\right)}}\right)+(1-y_i)\cdot\log\left(1-\frac{1}{1+e^{-\left(\beta_0+\sum_{j=1}^{p}\beta_j x_{ij}\right)}}\right)\right]+\lambda\sum_{j=1}^{p}\left\|\beta_j\right\|$$

where λ is the penalty factor to control for the penalty level. The higher the level is, the stricter the selection of critical factors is. The penalty term is composed of the sum of absolute values of coefficients. Therefore, factors with marginal influences would get zero coefficients in order to minimize the loss function.[2] The statistically-grounded Lasso regularization is applicable to a wide variety of supervised learning algorithms and has gained increasing popularity because of its ability to mitigate over-fitting and identify influential predictors in high-dimensional models (Qiao, Wang, & Yang, 2019; Wang, Cai, Chang, & Zurada, 2017; Zhu, Rosset, Tibshirani, & Hastie, 2003).

For the low-dimensional BTYD modeling, in line with prior studies (McCarthy et al., 2016; Zhang, 2008) on a similar discrete-time setting we apply the BG/BB model to the non-contractual transaction data for repurchase predictions. In contrast to multiple features explored by machine learning, the only required input information for BG/BB is a customer's binary transaction pattern illustrated in Fig. 3. In each period, a binary indicator is applied to show whether a customer has at least one transaction. Taking a quarter as a period, the first customer has the same purchase pattern (1 1 1 1) as that of the second customer. Yet, their purchase patterns will be quite different if we measure binary indicators over months. To better capture customer heterogeneities and increase variance in transaction patterns, we use monthly data for the BG/BB model estimation.[3] The key role BG/BB serves is simply generating outputs that are turned into quarterly features for all the other learning algorithms. The same logic applies to many other features that are also transformations/aggregations of monthly data points.

Specifically, we observe a customer's purchase pattern in terms of the number of periods for possible purchases (n: transaction opportunity), the number of periods with purchases occurred (x: frequency), and the most recent period with purchases occurred (t_x: the last active period, equivalent to current period index minus recency). In Fig. 3, for example, both the customers have the same $n=12$ transaction opportunities, i.e., the number of periods from the beginning till the time of analysis. Accordingly, the first customer shows the pattern of $x=12$ and t_x=12 while the second presents $x=4$ and $t_x=10$. In fact, the simple metrics (x, t_x) are sufficient statistics for the BG/BB model, illuminating the low-dimensional nature of this probability modeling approach.

The probabilistic assumptions of the BG/BB model allow us to use above simple inputs of purchase patterns to estimate how a customer would be alive for some periods and become inactive (death) afterwards. First, for each transaction period/opportunity, a customer has the probability θ to become inactive. Therefore, over periods, a customer turns into the inactive status in the ith trial, which can be described by the geometric distribution (dropout process). Before the ith trial that a customer becomes inactive, a customer's purchases can be described by a Bernoulli distribution with the purchased probability p for each opportunity over (i-1) trials (transaction process). Finally, customers are heterogeneous in terms of their purchase patterns: x, t_x, n. Two beta distributions are used to describe customer heterogeneities in the purchase probability $p\sim$beta(α,β) and dropout probability $\theta\sim$beta(γ,δ), where $\alpha, \beta, \gamma, \delta$ are parameters of two beta distributions. The four probability distributions above and their mixtures constitute the BG/BB model.

Given the number of transaction opportunities n, frequency x, and the last active period t_x calculated from data (see the example above), we estimate the BG/BB model using a maximum likelihood approach. After estimating the four parameters ($\alpha, \beta, \gamma, \delta$), we can then obtain a customer's probability estimates, i.e., the

[2] While the logit loss function is a standard for binary response modeling, adding Lasso regularization in practice could make the loss function difficult to optimize when the number of inputs and samples are high. A widely-adopted alternative for binary prediction models is to label y as 1 and −1 and fit the model using least squares loss that is much easier to optimize (Chatla and Shmueli 2017). In our cases, we find that the Lasso logit loss occasionally reveal difficulty to converge and the easy-to-tackle least squares loss leads to stable parameters and decent performance. So we report Lasso regression results derived from the more well-behaved Lasso squared loss function.

[3] BG/BB that relies on only two sufficient statistics will be short of granular information if it is calibrated on quarterly data. Estimating BG/BB based on monthly recency and frequency counts leads to better model outputs and fits our research objective of exploring its complementarity to machine learning better.

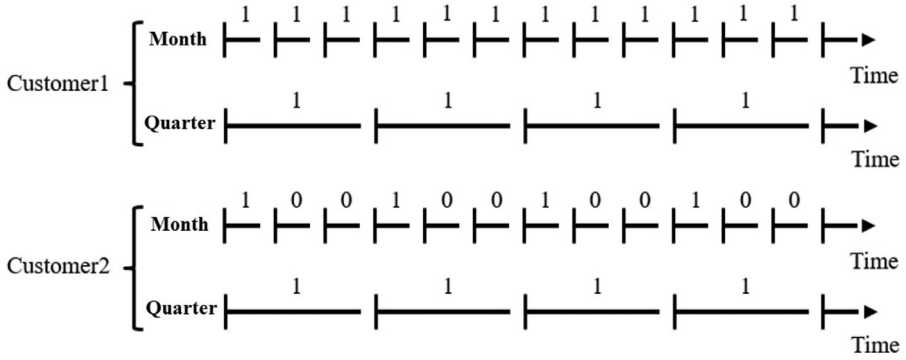

Fig. 3. Example of binary transaction pattern.

posterior mean transaction rate (E[p]), posterior mean dropout rate (E[θ]), and alive probability in the following transaction opportunity ($P(alive)$). In addition to the three probability estimates that reflect latent status of a customer, we further obtain the expected number of months with purchases over the future periods ($E[X(n, n+n^*)]$). This statistic explicitly captures a customer's purchase intensity, which can be transformed into whether he/she would purchase in the next quarter. For technical details and mathematical formulations of the log-likelihood function and probability estimates, please refer to Fader et al. (2010). Other than comparing prediction performance of Lasso from statistical machine learning and low-dimensional BG/BB from BTYD modeling, we aim to examine whether the two schools of modeling approaches can be complementary. We use the logit Lasso, to integrate the above four BG/BB estimates with high-dimensional features on behaviors related to cancel, return, delivery, payment, and discount.

4.2. Evaluation metrics and experimental settings

Since the target variable (Repurchase) is binary, any model should be evaluated under a threshold that transforms output probabilities into 0 and 1, where 0 stands for negative (no repurchase) and 1 stands for positive (repurchase)). Given the actual results, predictions can be classified as: true positive (TP), false positive (FP), true negative (TN), and false negative (FP). On the basis of the four classifications, different performance metrics have been proposed. According to Verbeke et al. (2012), performance of a classification model is usually evaluated in terms of area under the receiver-operating characteristic curve (AUC hereafter) and top-decile lift (TDL hereafter). These metrics are robust in that they evaluate model on various thresholds. In binary modeling, AUC quantifies the probability that a model ranks any randomly selected positive case higher than a randomly selected negative case, assuming positive ones have higher ranks (Burez & Van den Poel, 2009), and TDL describes how many times the focal model is more accurate than a random classifier when evaluating the two models on the top 10% of customers with the highest predicted probability of churn.

Although AUC and TDL are widely adopted, in cases with imbalanced targets, a popular alternative is the average precision score (APS hereafter). Similar to AUC that integrates true positive and false positive rates, APS is an overall performance measure for recall and precision over various thresholds, which is calculated as below:

$$\text{APS} = \sum_{j=1}^{h} (Recall_j - Recall_{j-1}) * Precision_j$$

where j stands for the threshold that recall (i.e., TP/(TP+FP)) and precision (i.e., TP/(TP+FN)) is computed. Thus, the idea of APS is a weighted average of precision at different threshold values using the increase in recall from the previous threshold as the weight. For customer repurchase prediction, having high recall and precision is desirable for the retailer that aims to maximize marketing efficacy while avoiding ineffective attempts/false positives. In addition, due to the nature of online platform, the distribution of our target variable is uneven (on average only 10–20% targets with $Repurchase = 1$). Essentially, the APS value stands for area under the precision-recall (PR) curve, which is suitable for skewed/imbalanced targets (Davis et al., 2005). Further, a model that dominates over the other in the precision-recall curve in principle would dominate the other in the ROC curve (Davis & Goadrich, 2006). Hence, in the following sections, we use APS in addition to AUC and TDL in order to obtain a holistic understanding of prediction performance.

After ensuring the methodological foundation of Lasso and selecting evaluation metrics, we have to design experimental settings. To match the operational requirements of the platform we work with, our model developed for predicting future repurchase requires *prospective validation* of all members, where the test set must be isolated from model tuning and *temporally forward* (Nelson et al., 2020). As a result, the panel data described in Section 3 is divided into a training set including the first 9 quarters and a *temporally forward* testing set based on the last quarter (Holtrop et al., 2017; Martínez et al., 2020). We train logistic regression with Lasso regularization (Lasso regression hereafter) on the training set, which is composed of the 92 input features extracted from members' transaction records from 1st to 9th quarter (see Fig. 2), and whether he/she would return in the following quarter (from 2nd to 10th) for repurchase as the target. To make optimization processes stable and effect estimates comparable, we standardize all input features using z-normalization. Models estimated from the training set are then applied to the test set for repurchase prediction in the 11th quarter (using the 92 features in the 10th quarter as the input features). The prediction is a continuous value within [0, 1], indicating the likelihood a customer would repurchase in the next quarter.

Like other machine learning methods, Lasso regression also has a hyper-parameter – the penalty value λ in the objective function – that affects its prediction performance. We use grid search and group 5-fold cross-validation (CV) (Pedregosa et al., 2011)[4] on the training set to determine the optimal value of λ. Unlike ordinary CV that randomly split samples into folds and fails to account for within-member dependencies over quarters, the group 5-fold CV

[4] In addition to group 5-fold CV, we use prequential validation with growing windows (Cui et al. 2018, Cerqueira et al. 2020) to optimize the hyper-parameter(s). Due to the large samples we have for model training, the identified optimal hyper-parameters are highly consistent for the two CV approaches.

ensures that samples from the same member would fall into the same fold, and hence better fits our empirical study on member-quarter panel data. Note that all of the other tested machine learning algorithms in the rest of our paper also follow the same group CV procedure for hyper-parameter optimization.

We estimate the BG/BB model based on monthly data of binary patterns explained earlier (see Fig. 3). We obtain customers' purchase patterns (x, t_x, n) from the training set: 1st to 30th months (1st to 10th quarters) and estimate associated parameters. Given these parameters, we compute $E[X(30,\ 30+3)]$ as the expected number of months that customers will return in the 11th quarter. That is, instead of generating repurchase probabilities in [0,1], the BG/BB model produces predictions from 0 up to 3 months to indicate how frequent customers may repurchase in the next quarter. To make the BG/BB predictions comparable to the output of Lasso regression, we scale $E[X(30,\ 30+3)]$ into [0, 1] (to approximate repurchase propensity of members) using min-max scaling.[5] As a result, we can evaluate prediction performance of machine learning and BTYD models for our binary repurchase problem.

After training and tuning proposed models, we evaluate all the models on the test set using the three afore-mentioned performance metrics – AUC, TDL, and APS. To statistically compare models, we evaluate each model 30 times on the *n-out-of-n bootstrap* sample of the temporally forward test set[6] (Dusenberry et al., 2020; Hughes et al., 2020; Klug et al., 2020; Rajkomar et al., 2018; Soffer, Klang, Barash, Grossman, & Zimlichman, 2020). The *n-out-of-n* test set bootstrap is an emerging practice for deriving test statistics under large samples and the test protocol is desirable when retraining a model many times is too costly or infeasible (Wood-Doughty, Shpitser, & Dredze, 2020). In line with related literature (De Bock & Van den Poel, 2011; Verbeke et al., 2011), we mainly use paired t-tests to compare model performance in terms of AUC, TDL, and APS. Three non-parametric tests are conducted for triangulation purposes. Because of the non-trivial number of models and metrics tested, we introduce the test method and present detailed test statistics and p-values for all model comparisons in Appendix D for the sake of brevity.

5. Empirical findings

5.1. Primary results

Table 3 shows the prediction performance of the test set (the 11th quarter) using different models, where bold numbers denote the best ones. The first row of Table 3 shows prediction performance of *BG/BB*, *Lasso-Current* that denotes Lasso regression with 65 features in the current quarter without historical information, and *Lasso-All* that denotes Lasso regression with all the 92 features in Fig. 2. We find that BG/BB, which only considers two sufficient statistics from past transaction, significantly outperforms Lasso-Current in terms of AUC (0.21%), TDL (2.18%) and APS (1.71%). Comparing BG/BB to *Lasso-All* with the opportunities to learn from detailed features on historical information, we find that *Lasso-All* significantly outperforms BG/BB in terms of AUC/APS by 0.35%/1.02%, but is significantly outperformed by BG/BB in TDL (0.58%). Our results indicate that high-dimensional Lasso regression does not necessarily outperform BG/BB that takes merely two inputs (R and F). We further explore how the two models can be complementary to each other. Specifically, we include three probability estimates (E[p], E[θ], P(alive)) and expected purchase intensity ($E[X(n, n+n^*)]$) from BG/BB as additional inputs to the Lasso regression. Accordingly, the second row of Table 3 shows performance of *Lasso-Current*(+*BG/BB*) with 69 features and *Lasso-All*(+*BG/BB*) with 96 features. The former improves *Lasso-Current* by 1.29%/5.47%/7.38% in AUC/TDL/APS, and the latter improves *Lasso-All* by 0.90%/3.78%/4.83% in AUC/TDL/APS. The improvements are all statistically significant.

Table 3
Summary of performance evaluation.

Model	AUC	TDL	APS
BGBB	0.774 (0.0012)	3.750 (0.0219)	0.323 (0.0028)
Lasso-Current	0.773 (0.0013)	3.670 (0.0164)	0.317 (0.0024)
Lasso-All	0.777 (0.0012)	3.728 (0.0204)	0.326 (0.0027)
Lasso-Current(+BGBB)	0.783 (0.0013)	**3.870 (0.0210)**	0.341 (0.0026)
Lasso-All(+BGBB)	**0.784 (0.0012)**	**3.870 (0.0198)**	**0.342 (0.0027)**

Standard deviations from 30 n-out-of-n bootstrap test set samples are in the parentheses.

Moreover, while Lasso regression without BG/BB estimates (in the first row) does not outperform the baseline BG/BB, incorporating BG/BB estimates into Lasso regression with full features, i.e., *Lasso-All(+BG/BB)*, leads to 1.26%, 3.18%, and 5.91% significant improvement in the three metrics over the BG/BB. As the four BG/BB estimates reflect customers' latent status and resulting purchase intensity, the finding indicates that these BG/BB outputs can add additional value to high-dimensional Lasso regression. Finally, as *Lasso-All* with historical information outperforms *Lasso-Current*, BG/BB complements *Lass-Current* and makes *Lasso-Current(+BG/BB)* with 69 features significantly outperforms *Lasso-All* with 92 features in AUC (0.73%), TDL (3.82%), and APS (4.51%). Additionally, the most complicated *Lasso-All(+BG/BB)* performs almost identical to *Lasso-Current(+BG/BB)* and differs by just 0.001 in AUC/APS, suggesting that BG/BB outputs extracted from timing patterns of past transactions might have already captured a valuable proportion of historical information.

As mentioned earlier, Lasso regression tends to shrink effect sizes of non-crucial variables and in our case, despite having ~70 to ~100 features in the four Lasso models in Table 3, most predictors turn out to have zero or close to zero coefficients. For each Lasso regression model, we identify five predictors with biggest absolute effect sizes and report the effects in Table 4. We report important features with their abbreviation, and add prefix 'Hist' to features derived from historical average. Note that the effect estimates are comparable among predictors because all features are already scaled by z-normalization before estimation. The first column shows that recency (R), the number of quarters since his/her first purchase (Age), two mean time between purchases indicators (BP-Normal and BP-Churn), and payment in convenience stores (PayInStores) are the most influential factors among 65 current quarter features. For the *Lasso-All* with historical information (92 features), standard deviation of time between purchases (STBP) and historical average of payment in convenience stores (Hist.PayInStores) outrank BP-Churn and PayInStores in terms of feature importance.

For *Lasso-Current(+BG/BB)* and *Lasso-All(+BG/BB)*, regardless of the inclusion of features on historical information, $E[X(n, n+n^*)]$ is the most influential predictor with positive effects. We observe that some influential features are outranked and shrunk

[5] Note that this may not be the best-case application of BG/BB predictions. For completeness, we fit a logistic regression where E[X(n, n+n*)] is the predictor and the Repurchase (1 or 0) is the response variable using training samples, such that E[X(n, n+n*)] can be scaled into [0, 1] in a model-based fashion. This, however, does not lead to performance gains so we report BG/BB results based on the easy-to-implement min-max scaling. Also, BG/BB tends to excel over longer forecast horizons (more than one quarter), as shown in related studies.

[6] A popular alternative for model testing is 5x2cv (Dietterich 1998), which is, however, usually deployed for cross-sectional data and computationally feasible only for non-large samples (Dietterich 1998). Also, our problem setting with more than 2.2 million training samples requires validation procedures that explicitly accommodate within-member observations and a temporally forward test set with full units. Hence, we adopt the n-out-of-n test set bootstrapping broadly used in recent machine learning studies.

Table 4
Important features in Lasso regression.

Lasso-Current		Lasso-All		Lasso-Current (+BGBB)		Lasso-All (+BGBB)	
Feature	Coef	Feature	Coef	Feature	Coef	Feature	Coef
R	−0.153	R	−0.129	E[X(n, $n+n^*$)]	0.129	E[X(n, $n+n^*$)]	0.115
BP-Normal	0.082	Age	0.096	R	−0.046	F	0.052
Age	0.081	BP-Normal	0.064	F	0.039	R	−0.050
PayInStores	0.072	STDB	−0.046	PayInStores	0.036	E[θ]	−0.028
BP-Churn	0.071	Hist.PayInStores	0.042	NetPay	0.029	Hist.NetPay	0.028

in their effect sizes due to the substitution effects brought by BG/BB outputs, i.e., $E[X(n, n+n^*)]$ and E[θ] (expected dropout rate) that is negatively related to repurchase likelihood. Incorporating BG/BB outputs into Lasso makes the influential BP-Normal in *Lasso-Current(+BG/BB)* as well as BP-Normal/STDB in *Lasso-All(+BG/BB)* fall out of top five influential predictors. On the other hand, we find that influential R and PayInStores in *Lasso-Current* are also revealed by *Lasso-Current(+BG/BB)*, but their effect sizes are greatly compressed (∼70% reduction in R and ∼50% reduction in PayInStores). That is, $E[X(n, n+n^*)]$ not only absorbs but also complements the information of current quarter features, resulting in better prediction performance. Comparing *Lasso-All(+BG/BB)* with *Lasso-Current(+BG/BB)*, we find that including historical information only mildly reduces the effect size of $E[X(n, n+n^*)]$, which still stands out and surpasses all other historical features. In both models, the effect size of $E[X(n, n+n^*)]$ is almost the sum of absolute effect sizes of other predictors in the same column. Also, R and F are consistently identified as important inputs and outrank other features such as payment in convenience stores (PayInStores), the net payment amount (NetPay), and the historical average of the net payment amount (Hist.NetPay). Together with the significant improvement in AUC/TDL/APS brought by BG/BB in Table 3, the outstanding effect of $E[X(n, n+n^*)]$ in Table 4 offers strong support for the complementarity between BG/BB and high-dimensional linear statistical machine learning.

5.2. Non-Linear machine learning algorithms

To consolidate our findings, we further relax the linear assumption behind the Lasso regression. We verify our results using feed-forward ANN that are more flexible and capable of capturing non-linear relationships. ANN are computational graphs composed of an input layer with all predictor variables and an output layer with target variable(s). One or more hidden layer(s) lies in between the input and output layers. Take a rudimentary case for illustration, a fully-connected ANN with 1 hidden layer and H hidden nodes, p input variables ($\boldsymbol{X}$), and one output node can be presented as

$$f^{ANN}(\boldsymbol{w}, \boldsymbol{X}) = g^o\left(\boldsymbol{b_{out}} + \sum_{h=1}^{H} \boldsymbol{w_h} g^h\left(\boldsymbol{b_h} + \sum_{j=1}^{p} w_{jh} X_{ij}\right)\right)$$

where $\boldsymbol{w}$ and $\boldsymbol{b}$ denote parameters to be learnt from data by minimizing a pre-specified loss function. The power of ANN comes from activation functions $g()$ that enable one to learn non-linear representations of input-output relationships. ANN have been widely used for cross-sectional study and recently have also been shown to be effective for panel data forecasting (Jahn, 2020). In Section 5.3, we will assess the robustness of apply our member-quarter ANN versus cross-sectional ANN, as well as sophisticated neural nets that explicitly consider sequential dependencies in longitudinal observations.

Following what we have done for Lasso modeling, we use the same training/testing data for ANN and determine optimal hyper-parameters (i.e., network topology, patience for early-stopping, and batch size) by grid search and group 5-fold cross-validation on the training data. Details regarding the architecture and optimization of ANN are reported in Appendix B. We report the results in Table 5 (where bold numbers denote the best ones) based on the same testing and naming convention in Table 3. *ANN-Current* uses only 65 input features without historical average in Fig. 2, whereas *ANN-All* uses all 92 features. These two neural nets are further augmented by the four BG/BB outputs, called *ANN-Current*(+*BG/BB*) and *ANN-All*(+*BG/BB*), respectively. Table 5 shows results from ANN that employ non-linear transformation to extract more information from data than Lasso regression. *ANN-Current* improves *Lasso-Current* by 1.10%/5.12%/5.61% in AUC/TDL/APS, whereas *ANN-All* improves *Lasso-All* by 0.97%/3.88%/4.87% in AUC/TDL/APS. The improvements are statistically significant.

Table 5
Summary of ANN performance.

Model	AUC	TDL	APS
ANN-Current	0.781 (0.0013)	3.858 (0.0203)	0.335 (0.0024)
ANN-Current(+BGBB)	0.783 (0.0013)	3.890 (0.0203)	0.342 (0.0026)
ANN-All	0.785 (0.0012)	3.873 (0.0207)	0.342 (0.0026)
ANN-All(+BGBB)	**0.785 (0.0012)**	**3.895 (0.0189)**	**0.345 (0.0026)**

Standard deviations from 30 n-out-of-n bootstrap test set samples are in the parentheses.

In cases where the BG/BB outputs are included, *ANN-Current*(+BG/BB) improves *ANN-Current* in AUC(0.22%)/TDL(0.84%)/APS(2.16%), whereas *ANN-All(+BG/BB)* improves AUC/TDL/APS of *ANN-All* by 0.08%/0.58%/0.87%. For both cases, the improvements are statistically significant. Consistent with Lasso regression results in Table 3, the BG/BB with its abilities to absorb R and F patterns of past transactions, complements ANN most in the case with only current quarter features. Even though ANN (*ANN-Current/All*) moderately outperforms Lasso regression without BG/BB estimates (*Lasso-Current/All*), the advantage of ANN is gone after BG/BB outputs are added to Lasso regression (*Lasso-Current/All(+BG/BB)*). In the presence of current quarter data and information provided by BG/BB, *ANN-Current(+BG/BB)* significantly improves *Lasso-Current(+BG/BB)* in AUC/TDL/APS by 0.04%/0.51%/0.48%, and *ANN-All(+BG/BB)* significantly improves *Lasso-All(+BG/BB)* in AUC/TDL/APS by 0.15%/0.68%/0.91%. A major takeaway is that if properly modeled, combining BG/BB estimates alongside other features in a regularized Lasso regression could perform almost equivalently to ANN but require much lower costs of model training and tuning.

In spite of observing that the inclusion of BG/BB outputs improves the performance of ANN, we would like to assess whether ANN performance could be compromised due to the inherent difficulty of optimizing model parameters and architecture. Therefore, we adopt a new type of ANN, which utilizes classification and regression tree (CART) (Breiman, Friedman, Stone, & Olshen, 1984) to identify top k important features from m raw features ($m>>k$), and then feed only the k features as inputs to ANN (Chakraborty, Chakraborty, & Murthy, 2019). In addition to establishing statistical properties (universal consistency under a single-hidden layer) of this CART-ANN modeling approach, Chakraborty et al. (2019) de-

Table 6
Important Features in CART-ANN.

CART-ANN-10 of Current(+BGBB)		CART-ANN-10 of All(+BGBB)	
Feature	Importance	Feature	Importance
E[X(n, $n+n^*$)]	0.1836	E[X(n, $n+n^*$)]	0.1145
Total Payment (Q)	0.1087	Total Payment (H)	0.1019
Net Payment (Q)	0.0801	Net Payment (H)	0.1011
Total Payment (A)	0.0758	Discount Amount (H)	0.0582
Net Payment (A)	0.0705	Total Payment (Q)	0.0448
Discount Amount (Q)	0.0361	Purchased Unit (H)	0.0324
Discount Amount (A)	0.0351	Holiday Transaction (H)	0.0317
E[p]	0.0236	Items Purchased (H)	0.0302
Purchased Unit (Q)	0.0195	Net Payment (Q)	0.0289
Purchased Unit (A)	0.0190	Coupon Usage (H)	0.0266

Q: current quarter; A: transactional average; H: historical average.

rive an expression[7] of the required hidden nodes. The analytical approach greatly reduces the burden of ANN architecture design as well as the number of total parameters to be optimized (due to reduced inputs).

We adopt the recently developed model and first train CART using current features and all features respectively. For each CART model, we identify $k=10$ important features based on Gini impurity and re-train corresponding ANN. CART-ANN with 10 of current inputs achieve 0.781/3.869/0.339 in AUC/TDL/APS, whilst CART-ANN with 10 of all features reach 0.784/3.865/0.340 in AUC/TDL/APS. Albeit greatly simplified, both ANN still perform close to their counterparts without feature selection (see Table 5).[8] More importantly, similar to feature importance revealed by Lasso in Table 4, Table 6 shows top 10 features identified by CART for both ANN. E[$X(n, n+n^*)$] is identified as the most influential feature in both models. However, R and F that emerge in *Lasso-Current(+BG/BB)* and *Lasso-All(+BG/BB)* models are not deemed as important features in CART-ANN. M-related (monetary) features such as amount of payment, discount, and purchase turn to be key inputs. In sum, the findings that E[$X(n, n+n^*)$] consistently emerges as the leading input feature, and that ANN with BG/BB reach the highest AUC/TDL/APS, consolidate our proposition: BGBB outputs can complement machine learning.

5.3. Robustness checks

The empirical tests in Sections 5.1 and 5.2 offer firm support for our proposition on the complementarity of BG/BB to linear Lasso and non-linear ANN for repurchase predictions. Unlike Lasso regression that has been proven to be consistent for longitudinal data (Medeiros & Mendes, 2016), ANN, as mentioned earlier, do not explicitly take into account time dependencies in our large N-small T panel. Despite recent empirical evidence for the efficacy of ANN for panel data forecasting (Jahn, 2020), we conduct robust checks and assess whether neural nets with sequential dependencies would outperform ordinary ANN. We employ *Recurrent Neural Networks* (RNN) (Goodfellow, Bengio, Courville, & Bengio, 2016), a class of neural nets that are capable of processing longitudinal data and connecting sequential feedback among input-output features. Specifically, we adopt Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) neural networks, both of which are state-of-the-arts sequence data modeling techniques. With the aid of input(s), output(s), and mathematical functions of forget gates as well as memory cells in hidden states for extracting sequential feedback information in each time step, LSTM has been shown to be effective for customer repurchase/churn prediction (Mena et al., 2019). GRU, on the other hand, has gained enormous popularity due to being simpler than LSTM (GRU only has update and reset gates), easier to train, and well-performing on many occasions (Cholet, 2017). We provide more explanations of LSTM and GRU in Appendix C. Goodfellow et al. (2016) also offer in-depth introduction to the two RNN. Although there are other regression or Markovian models for sequential data, we believe the sophistication and scalability of LSTM and GRU for large input dimensions makes them ideal candidates for robustness checks and strong benchmark for ANN and Lasso regression.

Table 7
Summary of model comparison.

Model	AUC	TDL	APS
Lasso-Current(+BGBB)	0.792 (0.0016)	4.384 (0.0205)	0.351 (0.0029)
Lasso-All(+BGBB)	**0.793 (0.0016)**	4.389 (0.0203)	0.352 (0.0029)
ANN-Current(+BGBB)	0.791 (0.0016)	4.401 (0.0244)	0.350 (0.0030)
ANN-All(+BGBB)	**0.793 (0.0016)**	**4.421 (0.0229)**	**0.353 (0.0030)**
LSTM	0.791 (0.0016)	4.399 (0.0246)	0.350 (0.0034)
GRU	0.789 (0.0016)	4.356 (0.0248)	0.341 (0.0034)

Standard deviations from 30 n-out-of-n bootstrap test set samples are in the parentheses.

Because LSTM and GRU that consider recursive feedback at each quarter require sequential observations as inputs for training, we limit our training samples to members with Age>=4 (2,069,459 member-quarter observations).[9] We limit the training/testing samples to the same used in the two RNN models and re-train Lasso regression and ANN for the sake of fair comparison. Following what has been done for Lasso, ANN, and CART-ANN in previous sections, grid search and group 5-fold CV on training data are used to determine the optimal value of hyper-parameters for LSTM and GRU. Details regarding the architecture and optimization of LSTM and GRU are reported in Appendix B. Table 7 shows prediction performance of four methods, where bold numbers denote the best ones. The proposed *Lasso-Current(+BG/BB)* significantly outperforms LSTM in AUC/APS by 0.01%/0.29%, and significantly outperforms GRU in all three metrics. On the other hand, ANN with BGBB delivers significantly higher AUC/TDL/APS than LSTM or GRU. The results suggest that the proposed Lasso-BGBB, with much lower training cost than complicated RNN, is able to capture relevant information from numerous members over quarters.

After empirically showing that Lasso-BGBB and ANN-BGBB could outperform RNN, we conduct additional tests to ensure that the two models mostly applied to cross-sectional data are not compromised by our sampling structures. Specifically, we re-train Lasso and BGBB models using (1) only cross-sectional observations in the ninth quarter, (2) 50–50 random under-sampling of the original panel data (preferred over-sampling according to Burez and Van den Poel (2009)), and (3) cross-sectional observations in the ninth quarter with 50–50 under-sampling. Then we compare the three sampling methods to the initial estimation approach using unbalanced panel data with all observations. The purpose of doing so is to assess the impact of longitudinal dependency and class imbalance (20% repurchase) on Lasso/ANN performance.

[7] Define n and d_m *as* the number of observations and the dimension of input layer in ANN, Chakraborty et al. (2019) derive $\sqrt{\frac{n}{d_m \log(n)}}$ as the number of hidden nodes robust to sample sizes.

[8] We have also identified the optimal number of retained features (48 for Current and 76 for All) through group 5-fold cross-validation. The CART-ANN with a lot more features only improves AUC by 0.02%, implying that our ANNs are not compromised and effective given only 10 features influential in CART.

[9] In general, members with fewer than q quarterly records are not predictable for RNN with time step q. A larger q implies more members are excluded for prediction. For $q=4$, there are ~25% members non-included, whereas for $q=8$, ~75% of members of the merchant are not predictable by RNN. After a discussion with the platform engineers, we set $q=4$ quarters that provide reasonable information for RNN while covering ~75% members (with Age>=4) for prediction assessment.

Table 8
Prediction performance under different sampling approaches.

Model	AUC	TDL	APS
Lasso-Current(+BGBB) full-panel	**0.783 (0.0013)**	**3.870 (0.0210)**	**0.341 (0.0026)**
Lasso-Current(+BGBB) cross-sectional	0.781 (0.0013)	3.868 (0.0219)	0.339 (0.0026)
Lasso-Current(+BGBB) under-sampling	0.781 (0.0013)	3.794 (0.0196)	0.334 (0.0026)
Lasso-Current(+BGBB) cross+under	0.780 (0.0013)	3.772 (0.0213)	0.329 (0.0026)
Lasso-All(+BGBB) full-panel	**0.784 (0.0012)**	**3.869 (0.0198)**	**0.342 (0.0027)**
Lasso-All(+BGBB) cross-sectional	0.782 (0.0013)	3.860 (0.0198)	0.340 (0.0027)
Lasso-All(+BGBB) under-sampling	0.782 (0.0013)	3.810 (0.0213)	0.332 (0.0029)
Lasso-All(+BGBB) cross+under	0.781 (0.0013)	3.791 (0.0218)	0.328 (0.0028)
ANN-Current(+BGBB) full-panel	**0.783 (0.0013)**	**3.890 (0.0203)**	**0.342 (0.0026)**
ANN-Current(+BGBB) cross-sectional	0.776 (0.0013)	3.829 (0.0203)	0.333 (0.0026)
ANN-Current(+BGBB) under-sampling	0.782 (0.0013)	3.873 (0.0203)	0.338 (0.0026)
ANN-Current(+BGBB) cross+under	0.780 (0.0013)	3.818 (0.0213)	0.330 (0.0026)
ANN-All(+BGBB) full-panel	**0.785 (0.0012)**	**3.895 (0.0189)**	**0.345 (0.0026)**
ANN-All(+BGBB) cross-sectional	0.777 (0.0012)	3.782 (0.0210)	0.333 (0.0025)
ANN-All(+BGBB) under-sampling	0.784 (0.0012)	3.891 (0.0211)	0.342 (0.0025)
ANN-All(+BGBB) cross+under	0.782 (0.0012)	3.806 (0.0219)	0.329 (0.0026)

Standard deviations from 30 n-out-of-n bootstrap test set samples are in the parentheses.

Table 8 reports testing results of four different sampling approaches, where bold numbers denote the best ones. Among all of the tested specifications – *Lasso-Current(+BG/BB)*, *Lasso-All(+BG/BB)*, *ANN-Current(+BG/BB)*, and *ANN-All(+BG/BB)*, models trained from full panel data structures consistently deliver the best performance in AUC/TDL/APS. Using only cross-sectional training data in the most recent quarter in attempt to conform with independent sampling properties, however, does not improve prediction performance. Under-sampling does not outperform its imbalanced counterpart either, indicating our mildly imbalanced targets do not compromise ordinary modeling performance. Finally, for both Lasso and ANN models, using cross-sectional data with under-sampling consistently results in the worst performance. Taken together, in line with econometric studies on Lasso regression (Medeiros & Mendes, 2016; Smeekes & Wijler, 2018) and ANN (Jahn, 2020), the robustness checks suggest that it is desirable for the two methods to leverage information embedded in large member-quarter data sets. The empirical results support our use of full panel for model training.

6. Discussion and conclusions

Models for customer repurchase/churn prediction, particularly in non-contractual settings, have drawn wide attention from researchers in marketing, operations research, computer science, and so on. Our study contributes to the literature by analyzing an under-investigated link between BTYD models based on low-dimensional statistics and data mining/machine learning approaches to customer base analysis. The divide between the two methodological streams might be attributed to the extent to which the researcher is willing to shift between the two edges of causality (i.e., structural formulation) and correlation (i.e., brute-force function approximation). Or it might be due to analysts' intention to leverage an array of features derived from cross-sectional data as opposed to R and F statistics accumulated over time. Intrigued by the divide in the literature, we empirically compare the performance of BG/BB to Lasso and ANN. We find that notwithstanding the seemingly restrictive probabilistic assumptions, the BG/BB model performs reasonably well. In our study, the BG/BB model with understandable data-generating assumptions stands as a strong option for customer predictive analytics.

Moreover, we show the predicted future transaction of BG/BB, when used as an input to Lasso regression, is the most influential predictor for the model to further improve prediction accuracy. This hybrid approach – jointly using BTYD outputs and other features in machine learning – is simple and to some extent addresses the potential shortcoming that BTYD models usually consider limited covariates. While it may be technically possible to integrate additional covariates into BTYD models, doing so would just result in complex parameterizations that make the already complicated log-likelihood functions more difficult to optimize. The fact that BTYD models rarely leverage other covariates (such as categorical data of customer profile) may be the main reason why BTYD models are rarely introduced as a benchmark to machine learning experiments in the literature of customer scoring. Our findings suggest that researchers should not view machine learning as default alternatives to BTYD models for customer repurchase/churn prediction. Following our contribution to the literature of predictive analytics, we suggest researchers in marketing and data analytics to continue investigating the intersection of probabilistic BTYD models and machine learning algorithms.

In addition to the afore-mentioned implications for researchers, our modeling effort has non-trivial implications for practitioners. Even though the context of our study is within online retailing, our prediction problem is also relevant to other industry sectors. How to efficiently calibrate customer repurchase/churn prediction models using historical transactions is crucial for platforms as well as individual retailers. For one thing, instead of high-dimensional features from historical records, our predictive modeling approach only requires features from current transaction records and BG/BB estimates. Estimating the BG/BB model demands only simple inputs of transaction timing, and can be easily computed in even Excel spreadsheets. Comparing to massive development of machine learning approaches to customer base analysis, low-dimensional BTYD models still have broad applications in the real world because of their underlying simplicity and training efficiency (Gauthier, 2017). For another, provided the critical input feature from the BG/BB model, the prediction performance of an integrated Lasso regression is comparable to that of black-box models allowing for non-linear interactions in empirical tests. The hybrid regression modeling approach can be useful to practitioners who prefer transparent models and interpretable predictions. This is in line with the recent research trend that seeks for higher interpretability of machine learning models. Last, in practice, neither all members are predictable, nor they all have to be predicted for marketing campaign considerations. In fact, managers of the service provider posit that retailers on the platform usually only target at a subset of their customers when launching

marketing initiatives in a new quarter. Preliminary tests in our research site suggest that splitting members into RFM-based groups helps managers identify groups with better predictability. For practical operations, our Lasso-BG/BB model can be used in conjunction with data-driven decision rules for identifying predictable segments such that marketing campaigns can be more cost-effective.

Our study has several limitations that leave opportunities for future research. First, given the discrete, non-contractual problem setting, we focus on the BG/BB model and empirically investigate its complementarity to machine learning algorithms. Continuous and/or contractual data are, however, beyond the scope of our study. Subsequent studies on customer repurchase prediction should explore the remaining categories. On top of the binary repurchase incidence commonly assessed in our paper and prior studies (e.g., Chou & Chuang, 2018, Martínez et al., 2020, Suh et al., 2004), subsequent studies can further predict continuous metrics such as repurchase dollars/quantities. Second, our empirical analysis is based on data from one online apparel retailer, thus limiting the generalizability of our results. That said, the operational setting (i.e., a web-based specialty retailer) and available customer information (i.e., purchase/payment/return activities) are nearly identical to many other online retailers that also intend to predict customer intention based on historical transactions. Hence, subsequent studies can follow our analysis protocol and triangulate our findings. Third, our transactional features do not include customer website browsing activities. Our retailer runs its business on a platform enabled by a technological service provider and the service provider is not allowed to release customers' detailed clickstreams in neither the focal retailer nor other retailers who use the same platform services. While cross-retailer clickstream data of each customer might help machine learning algorithms improve prediction performance, such data are unavailable to many retailers and involve privacy issues to be resolved. When data is available, future studies can explore how to incorporate unstructured information such as customers' clickstream behaviors and social network.

Despite the research limitations, our study aims to be a provoking example for researchers in marketing modeling and machine learning. Both schools of researchers develop numerous models for the sake of using historical data in prediction tasks related to customer purchase, churn, and value. However, in the academic research community, it seems that there exists a dichotomy between low-dimensional BTYD models in marketing and learning algorithms utilizing high-dimensional inputs from computer science, statistics, and operations research. On the contrary, practitioners have already begun investigating BTYD modeling versus machine learning approaches for customer base analysis (Google Cloud 2019). Data scientists in the industry (Brownell, 2019) also report evidence that various continuous, non-contractual BTYD models outperform RNN in predicting future customer transactions. In responses to practitioners' interests in comparing and contrasting the two customer modeling avenues, we posit that the divide in the literature should be broken and encourage more researchers to contribute to the emerging literature on cross-fertilizing BTYD and machine learning.

Appendix A. Transaction data tables and pre-processing processes

In Table A.1, we present the raw transaction data obtained from the online service provider. The dataset is composed of three inter-related tables. The *Purchase* table provides records of each customer order containing specifics on member id, cart id, and product id. In addition to the three identifiers, the *Purchase* table contains a variety of attributes related to this order, including price, quantity, device, channel, payment, delivery, etc. After an order is placed, it is possible for a member to cancel or return purchased goods later on. The *Cancel* and *Return* tables document those events. Note that members are allowed to partially cancel or return a subset of their purchase. The tables also contain details on cancel/return time, product, quantity, reason, etc. Both *Cancel* and *Return* have an attribute *detail id* being an identifier for each cart-product pair, such that we can connect cancel/return events to specific orders. The three tables jointly constitute an episode of one customer transaction. Note that the service provider screens for outlying observations within its site and asks us to leverage all raw data.

Table A.1
Overview of the raw dataset.

Table	Columns	Description	Type
Purchase	Member Id	The identifier of member	String
	Cart Id	The identifier of cart	String
	Product Id	The identifier of product	String
	Detail Id	The identifier of (Cart Id, Product Id) pair	String
	Timestamp	The date and time of transaction	Date-time
	Category	The product category and sub-category	String
	Is Major	The indicator of major product	Binary
	Is Gift	The indicator of gift	Binary
	Price	The unit price	Integer
	Quantity	The purchased quantity	Integer
	Discount	The promotional discount value	Integer
	Device	The used device (Mobile, PC)	Dummy
	Source	The used channel (Android, iOS, Web)	Dummy
	Payment	Price * Volume – Discount – Coupon Discount	Integer
	Payment Type	The payment tool (Cash, Credit Card, ATM)	Dummy
	Coupon Id	The identifier of e-coupon	String
	Coupon Discount	The discount value of e-coupon	Integer
	Pickup Type	The approach for receiving goods	Dummy
Cancel	Detail Id	The identifier of (Cart Id, Product Id) pair	String
	Is Cancel	The indicator of cancellation	Binary
	Cancel Refund	The refund amount of cancel	Integer
	Cancel Quantity	The quantity of cancel	Integer
	Cancel Time	The cancelation time	Date-time
	Cancel Reason	The reason of cancelation	Dummy
Return	Detail Id	The identifier of (Cart Id, Product Id) pair	String
	Is Return	The indicator of returned transaction	Binary
	Return Refund	The refund amount of return	Integer
	Return Quantity	The quantity of return	Integer
	Return Time	The return time	Date-time
	Return Reason	The reason of return	Dummy

After we join the three tables based on the common product id and cart id for each transaction, we compute transaction details of each member id within each quarter of 3 months, in line with operational needs for quarterly analyses. The transaction details are then aggregated into summary statistics that characterize member behavior in a quarter (x_1 to x_{28} in Table 2). After repeating this process across all members and quarters, we obtain an unbalanced panel data of $N = 496{,}536$ members by 10 quarters. Given the data structure with 28 quarterly features, we go on to compute features transaction average, historical average, Age, and quarter fixed effects terms shown in Fig. 2. Note that the resulting panel data structure has no missing values. Since there are a total of ~100 features, for brevity we report descriptive statistics of the rudimentary 28 base features in Table A.2.

Table A.2
Descriptive statistics ($n = 2707{,}356$).

	Mean	Std.	Min	Max		Mean	Std.	Min	Max
x_1	7.057	6.225	0	27	x_{15}	−9.302	31.996	−200	0
x_2	0.369	0.645	0	3	x_{16}	0.025	0.177	0	3
x_3	0.117	0.354	0	2	x_{17}	0.065	0.284	0	3
x_4	1.454	2.898	0	17	x_{18}	0.059	0.285	0	4
x_5	1.473	2.929	0	17	x_{19}	0.115	0.648	0	6
x_6	333.201	660.244	0	4020	x_{20}	25.614	143.541	0	1395
x_7	0.027	0.197	0	3	x_{21}	0.023	0.160	0	2
x_8	0.000	0.016	0	1	x_{22}	0.027	0.249	0	3
x_9	0.027	0.196	0	3	x_{23}	5.508	54.640	0	749
x_{10}	0.379	0.742	0	7	x_{24}	272.965	560.570	0	3306
x_{11}	0.343	0.627	0	3	x_{25}	0.060	0.390	0	4
x_{12}	0.558	1.674	0	11	x_{26}	0.026	0.245	0	3
x_{13}	−30.669	93.316	−654	0	x_{27}	0.038	0.323	0	4
x_{14}	0.373	1.321	0	9	x_{28}	0.024	0.234	0	3

Appendix B. Hyper-parameters and architecture of ANN, LSTM, and GRU

We only consider ANN with at most 2 hidden layers (Ładyżyński, Żbikowski, & Gawrysiak, 2019; Martínez et al., 2020; Yu et al., 2018), and cross-validate hidden nodes from 10 to 90, by increments of 20, in addition to limiting the nodes in the second hidden layer to be less than or equal to the nodes in the first hidden layer. For notational brevity, we denote 1-layered ANN with N_1 nodes as (N_1), and 2-layered ANN with N_1 and N_2 nodes as (N_1, N_2). We set *epoch* to a large number and optimize *patience* (i.e., *patience*+1 is the consecutive number of allowed unimproved training iterations) to dynamically terminate the training process. Owing to poor convergence of mini-batch in our pre-test, the batch size is set to be proportional to the sample size of the training set, as shown in the rightmost column. We use Adam in Tensorflow to train ANN and set other optimizer-related parameters according to the recommendation by Kingma and Ba (2014). As described in the paper, the best hyper-parameters are identified by the group 5-fold CV of all training samples.

As for LSTM and GRU, we set the number of sequential time blocks to 4 quarters and each block learns from 65 features (used in Lasso-Current and ANN-current) that depict customer behavior in the corresponding quarter. Because LSTM and GRU by construction would extract historical information over time, hand-crafted features on past transactions are not needed for each block that carries over information from quarter to quarter. We adopt a 1 hidden-layer structure following related work (Mena et al., 2019; Salehinejad & Rahnamayan, 2016). The number of hidden nodes are cross-validated from 10 to 190, with increments of 10. Table B.2 shows hyper-parameters to be optimized. Same as Lasso and ANN, we identify the best setting using the group 5-fold CV of training samples.

Table B.1
Parameter setting for experiments with ANN.

Layers	Hidden Nodes	Patience	Batch Fraction
1, *2*	(90), (70), (50), (30), (10) (90, 90), ..., (50, 30), ..., (*30, 10*), ..., (10, 10)	[*1*, 2, 3]	[*1e-3*, 2e-3, 5e-3]

italic: selected parameters for ANN-Current; underline: selected parameters for ANN-All.

Table B.2
Parameter setting for experiments with LSTM and GRU.

	Layer	Hidden Nodes	Patience	Batch Fraction
LSTM	1	[10, 30, 50, 70, 90, 110, 130, 150, 190]	[1, 2, 3, 4]	[1e-2, 2e-2, 5e-2]
GRU	1	[10, 30, 50, 70, 90, 110, 130, 150, 190]	[1, 2, 3, 4]	[1e-2, 2e-2, 5e-2]

underline: the selected parameters.

Appendix C. LSTM and GRU

The core idea of RNN is that, instead of taking all the variables collected at the same period as inputs like feedforward ANN do, RNN take (multi-variate) time series of realized observations as inputs to generate predictions. RNN embed a sequential structure looking back to n periods ago. In each period/block, current states X_t together with previous hidden states h_{t-1} are used to generate current hidden states h_t. The rudimentary RNN, however, capture only short-term dependencies of time series. Therefore, LSTM and GRU, two powerful and popular variants of RNN, have been developed to capture both short-term and long-term dependencies. Fig. C.1 illustrates the recurrent structure of LSTM (in the top half) and GRU (in the bottom half). LSTM stores short-term and long-term dependencies in two distinct hidden states (S and L), whereas GRU captures both dependencies in a single hidden state (H). Same as RNN, both models generate and update the dependencies simply using current states X_t together with previous hidden states (S_{t-1} and L_{t-1} for LSTM, and H_{t-1} for GRU).

The major difference between LSTM and GRU lies in the mathematical formulations of their hidden states, i.e., the LSTM and GRU cells in Fig. C.1. In Figs. C.2 and C.3, we present the computation structures of LSTM and GRU cells in details. Define the gates (i.e., square blocks) as activation functions; symbols $\oplus$ and $\otimes$ as sum and product operations; in-flow/out-flow arrows as input/output. All inputs are turned to weighted sum before being fed into activation functions. For instance, when X_t and $S_{(t-1)}$ (or $H_{(t-1)}$ for GRU) are simultaneously fed into sigmoid activation function (denoted by $\sigma()$), the output is calculated as $\sigma(W_X X_t + W_h S_{(t-1)})$, in which W_X and W_h are weights to optimize. For LSTM cells shown in Fig. C.2, the two states S_t and L_t are generated from three gates – *Forget Gate*, *Input Gate*, and *Output Gate* –store short-term and long-term information separately in order capture dependencies among current and previous hidden states over time. As for GRU cells in Fig. C.3, the hidden states H_t with simplified design to improve computational efficiency are produced by two gates, i.e., *Reset Gate* and *Update Gate*. Albeit with fewer gates, GRU is still capable of capturing sequential patterns in short and long runs.

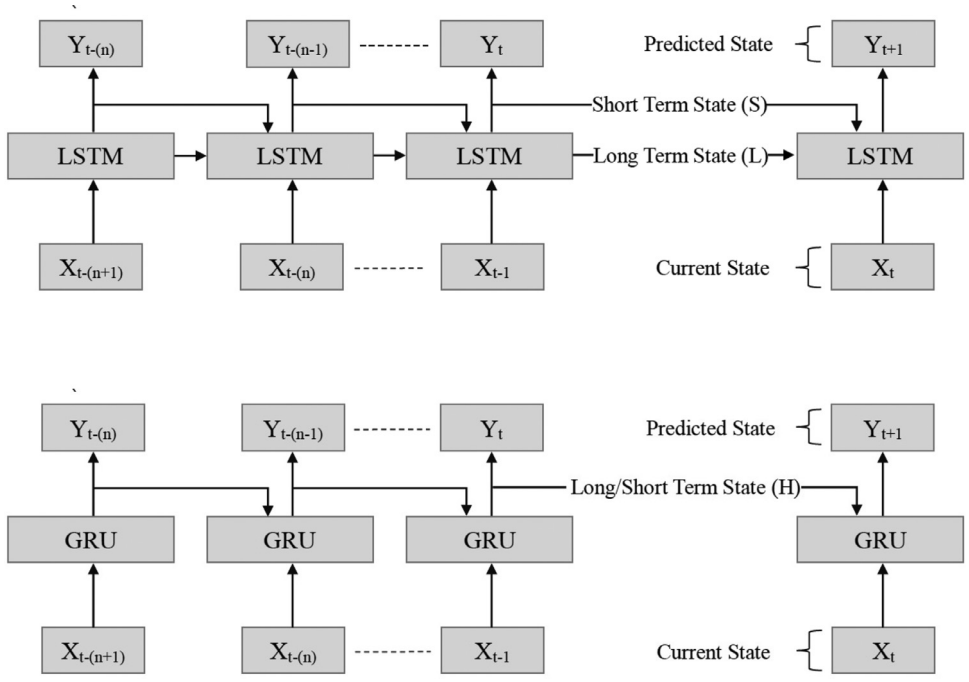

Fig. C.1. Recurrent structure of LSTM/GRU.

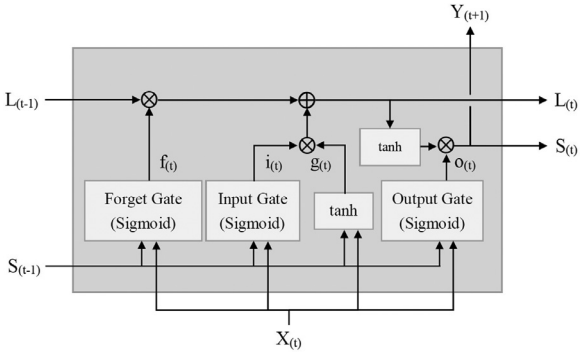

Fig. C.2. Complete structure of LSTM cell.

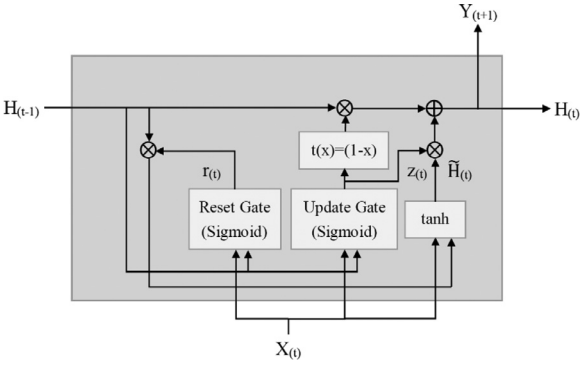

Fig. C.3. Complete structure of GRU cell.

Appendix D. Statistical test results of model comparison

For the various models in the paper, we evaluate each model 30 times on the *n-out-of-n bootstrap* sample of the temporally forward test set (Dusenberry et al., 2020; Hughes et al., 2020; Klug et al., 2020; Rajkomar et al., 2018; Soffer et al., 2020). Using the 30 samples on three performance metrics, we statistically compare models based on the paired *t*-test (Dietterich, 1998).

Denote A and B as two algorithms to be compared, N as the number of n-out-of-n bootstrapped subsets of the test set, $P^A = [P_1^A, P_2^A, \ldots, P_N^A]$ and $P^B = [P_1^B, P_2^B, \ldots, P_N^B]$ as the performance measures of A and B on the N subsets. Let $P_i = (P_i^A - P_i^B)$ be the performance difference of A and B on the ith subset, the paired t-test calculates the following t statistic the null hypothesis that the mean difference $\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$ is equal to zero (i.e., A and B are not statistically different):

$$t = \frac{\bar{P} \cdot \sqrt{N}}{\sqrt{\frac{\sum_{i=1}^{N}(P_i - \bar{P})^2}{N-1}}}$$

A large positive t score would suggest that A outperforms B, whereas a more negative t score would indicate the reverse. The following tables present the results of pair-wised comparisons based on the paired t-test. Table D.1 shows the statistical comparison of models in Tables 3 and 4 of the paper; Tables D.2 and D.3 separately show the comparison of models in Tables 7 and 8 of the main text. The values in the tables are the t statistics, and the star signs are the significance level of the comparisons. A positive t means the model in the row outperforms the model in the column, and a negative t refers to the opposite. In addition to paired t-test, we also confirm the testing results based on Wilcoxon sign rank test (Coussement & De Bock, 2013; Zhu, Baesens, & Van Den Broucke, 2017), a non-parametric alternative of paired t-test (Demšar, 2006), as well as Friedman test with post-hoc Holm test (De Bock & Van den Poel, 2012; De Caigny et al., 2018). The testing result of these approaches are highly consistent. Because of the non-trivial number of models and metrics, we only retain the results of the paired t-test.

Table D.1
Paired *t*-test of primary result.

		1 BG/BB	2 Lasso Current	3 Lasso All	4 Lasso Current (+BG/BB)	5 Lasso All (+BG/BB)	6 ANN Current	7 ANN All	8 ANN Current (+BG/BB)	9 ANN All (+BG/BB)
AUC	2	−19.06 ***								
	3	33.66 ***	86.40 ***							
	4	156.13 ***	177.16 ***	94.63 ***						
	5	156.13 ***	199.48 ***	180.35 ***	44.23 ***					
	6	111.70 ***	168.50 ***	71.39 ***	−32.55 ***	−54.62 ***				
	7	144.12 ***	187.77 ***	194.97 ***	37.27 ***	15.68 ***	63.29 ***			
	8	160.67 ***	221.04 ***	115.77 ***	11.01 ***	−29.83 ***	66.45 ***	−35.77 ***		
	9	148.89 ***	202.77 ***	186.56 ***	56.00 ***	40.75 ***	68.68 ***	29.07 ***	48.11 ***	
TDL	2	−28.58 ***								
	3	−8.54 ***	30.23 ***							
	4	48.22 ***	90.68 ***	63.86 ***						
	5	48.11 ***	106.92 ***	71.54 ***	−1.42					
	6	41.63 ***	82.74 ***	56.58 ***	−6.99 ***	−6.79 ***				
	7	51.51 ***	73.69 ***	51.16 ***	1.10	1.82 *	7.35 ***			
	8	60.55 ***	114.42 ***	71.62 ***	16.21 ***	15.53 ***	17.51 ***	8.72 ***		
	9	67.18 ***	105.08 ***	68.89 ***	14.35 ***	16.15 ***	23.81 ***	16.50 ***	3.55 ***	
APS	2	−20.46 ***								
	3	15.45 ***	70.89 ***							
	4	116.57 ***	154.94 ***	127.95 ***						
	5	129.13 ***	151.81 ***	151.82 ***	28.89 ***					
	6	59.02 ***	138.26 ***	72.94 ***	−53.54 ***	−57.67 ***				
	7	109.37 ***	151.07 ***	152.89 ***	13.33 ***	1.55	72.09 ***			
	8	126.11 ***	159.81 ***	137.54 ***	31.14 ***	8.70 ***	83.21 ***	4.90 ***		
	9	122.64 ***	179.61 ***	158.97 ***	53.50 ***	41.66 ***	107.61 ***	43.71 ***	31.45 ***	

H0: $\overline{P^{row} - P^{column}} = 0$, where P^{row}/P^{column} are the performance of model in the row/column.
*significant at $p < 0.05$; **significant at $p < 0.01$; ***significant at $p < 0.001$.

Table D.2
Paired *t*-test of robustness check with RNN.

		1 Lasso Current (+BG/BB)	2 Lasso All (+BG/BB)	3 ANN Current (+BG/BB)	4 ANN All (+BG/BB)	5 LSTM	6 GRU
AUC	2	49.42 ***					
	3	−3.62 ***	−46.14 ***				
	4	32.04 ***	6.03 ***	37.39 ***			
	5	−1.91 *	−28.64 ***	0.13	−30.46 ***		
	6	−42.71 ***	−66.85 ***	−42.76 ***	−58.41 ***	−59.04 ***	
TDL	2	3.56 ***					
	3	9.44 ***	5.57 ***				
	4	20.41 ***	17.69 ***	8.59 ***			
	5	5.92 ***	3.27 **	−0.91	−7.19 ***		
	6	−10.51 ***	−11.12 ***	−15.59 ***	−20.51 ***	−17.87 ***	
APS	2	26.34 ***					
	3	−6.14 ***	−23.61 ***				
	4	18.05 ***	8.73 ***	29.80 ***			
	5	−5.12 ***	−12.46 ***	−2.49 **	−18.14 ***		
	6	−41.15 ***	−52.37 ***	−41.11 ***	−63.44 ***	−60.75 ***	

H0: $\overline{P^{row} - P^{column}} = 0$, where P^{row}/P^{column} are the performance of model in the row/column.
* significant at $p < 0.05$; ** significant at $p < 0.01$; *** significant at $p < 0.001$.

Table D.3
Paired *t*-test of robustness check on Lasso and ANN.

			Current				All			
			1 Panel	2 Cross-Sectional	3 Under- Sampling	4 Cross+ Under	1 Panel	2 Cross-Sectional	3 Under- Sampling	4 Cross+ Under
Lasso	AUC	2	−50.86 ***				−49.12 ***			
		3	−58.30 ***	−0.55			−77.93 ***	−3.34 **		
		4	−81.45 ***	−19.05 ***	−88.37 ***		−90.38 ***	−20.05 ***	−56.72 ***	
	TDL	2	−2.16 **				−7.54 ***			
		3	−44.26 ***	−40.42 ***			−39.77 ***	−27.84 ***		
		4	−50.67 ***	−49.38 ***	−16.94 ***		−47.28 ***	−39.14 ***	−10.68 ***	
	APS	2	−38.52 ***				−34.92 ***			
		3	−84.82 ***	−53.97 ***			−95.57 ***	−65.52 ***		
		4	−91.22 ***	−81.64 ***	−78.70 ***		−92.74 ***	−80.21 ***	−47.08 ***	
ANN	AUC	2	−116.72 ***				−140.48 ***			
		3	−64.66 ***	99.97 ***			−40.60 ***	108.47 ***		
		4	−78.20 ***	79.36 ***	−37.99 ***		−93.18 ***	86.18 ***	−59.10 ***	
	TDL	2	−32.79 ***				−60.17 ***			
		3	−14.45 ***	21.81 ***			−3.08 **	43.40 ***		
		4	−37.97 ***	−5.12 ***	−25.67 ***		−47.94 ***	10.92 ***	−39.66 ***	
	APS	2	−81.46 ***				−114.42 ***			
		3	−52.90 ***	46.27 ***			−38.81 ***	83.72 ***		
		4	−80.61 ***	−17.74 ***	−49.01 ***		−129.81 ***	−24.79 ***	−112.81 ***	

H0: $\overline{P^{row} - P^{column}} = 0$, where P^{row}/P^{column} are the performance of model in the row/column.
*significant at $p < 0.05$; ** significant at $p < 0.01$; ***significant at $p < 0.001$.

References

Abe, M. (2009). Counting your customers" one by one: A hierarchical Bayes extension to the Pareto/NBD model. *Marketing Science, 28*(3), 541–553.

Alboukaey, N., Joukhadar, A., & Ghneim, N. (2020). Dynamic behavior based churn prediction in mobile telecom. *Expert Systems with Applications, 162*, Article 113779.

Baesens, B., Viaene, S., Van den Poel, D., Vanthienen, J., & Dedene, G. (2002). Bayesian neural network learning for repeat purchase modelling in direct marketing. *European Journal of Operational Research, 138*(1), 191–211.

Batislam, E. P., Denizel, M., & Filiztekin, A. (2007). Empirical validation and comparison of models for customer base analysis. *International Journal of Research in Marketing, 24*(3), 201–209.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brownell, A. (2019). *RNNS vs traditional ML for predictive customer lifetime value*. Available at https://retina.ai/blog/rnns-vs-traditional-ml/

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. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36*(3), 4626–4636.

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning, 109*(11), 1997–2028.

Chakraborty, T., Chakraborty, A. K., & Murthy, C. A. (2019). A nonparametric ensemble binary classifier and its statistical properties. *Statistics & Probability Letters, 149*, 16–23.

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 158*(3), 419–444.

Chatla, S. B., & Shmueli, G. (2017). An extensive examination of regression models with a binary outcome variable. *Journal of the Association for Information Systems, 18*(4), 1.

Chen, P. P., Guitart, A., del Río, A. F., & Periánez, A. (2018). Customer lifetime value in video games using deep learning and parametric models. In 2018 IEEE international conference on Big Data (Big Data) *(pp. 2134-2140)*. IEEE.

Cholet, F. (2017). *Deep learning with python*. Manning Publications.

Chou, Y. C., & Chuang, H. H. C. (2018). A predictive investigation of first-time customer retention in online reservation services. *Service Business, 12*(4), 685–699.

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.

Cui, H., Rajagopalan, S., & Ward, A. R. (2020). Predicting product return volume using machine learning methods. *European Journal of Operational Research, 281*(3), 612–627.

Cui, R., Gallino, S., Moreno, A., & Zhang, D. J. (2018). The operational value of social media information. *Production and Operations Management, 27*(10), 1749–1769.

Davis, J., Burnside, E. S., de Castro Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S., et al. (2005). View learning for statistical relational learning: With an application to mammography. *IJCAI*, 677–683.

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on machine learning* (pp. 233–240).

De Bock, K. W., & Van den Poel, D. (2011). An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. *Expert Systems with Applications, 38*(10), 12293–12301.

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*(Jan), 1–30.

Dew, R., & Ansari, A. (2018). Bayesian nonparametric customer base analysis with model-based visualizations. *Marketing Science, 37*(2), 216–235.

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

Dusenberry, M. W., Tran, D., Choi, E., Kemp, J., Nixon, J., Jerfel, G., et al. (2020,). Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM conference on health, inference, and learning* (pp. 204–213).

Fader, P. S., & Hardie, B. G. (2009). Probability models for customer-base analysis. *Journal of Interactive Marketing, 23*(1), 61–69.

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). Counting your customers" the easy way: An alternative to the Pareto/NBD model. *Marketing Science, 24*(2), 275–284.

Fader, P. S., Hardie, B. G., & Shang, J. (2010). Customer-base analysis in a discrete–time noncontractual setting. *Marketing Science, 29*(6), 1086–1108.

Gauthier, J. (2017). *An introduction to predictive customer lifetime value modeling*. Available at https://blogs.oracle.com/datascience/an-introduction-to-predictive-customer-lifetime-value-modeling

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*. Cambridge: MIT Press.

Google Cloud (2019). *Predicting customer lifetime value with AI platform: Introduction*. Available at https://cloud.google.com/solutions/machine-learning/clv-prediction-with-offline-training-intro

Gopalakrishnan, A., Bradlow, E. T., & Fader, P. S. (2017). A cross-cohort changepoint model for customer-base analysis. *Marketing Science, 36*(2), 195–213.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., et al. (2006). Modeling customer lifetime value. *Journal of Service Research, 9*(2), 139–155.

Hadden, J., Tiwari, A., Roy, R., & Ruta, D. (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research, 34*(10), 2902–2917.

Holtrop, N., Wieringa, J. E., Gijsenberg, M. J., & Verhoef, P. C. (2017). No future without the past? Predicting churn in the face of customer privacy. *International Journal of Research in Marketing, 34*(1), 154–172.

Hopmann, J., & Thede, A. (2005). Applicability of customer churn forecasts in a non–contractual setting. *Innovations in classification, data science, and information systems* (pp. 330–337). Berlin, Heidelberg: Springer.

Hughes, M. C., Pradier, M. F., Ross, A. S., McCoy, T. H., Perlis, R. H., & Doshi-Velez, F. (2020). Assessment of a prediction model for antidepressant treatment stability using supervised topic models. *JAMA Network Open, 3*(5), Article e205308 e205308.

Jahn, M. (2020). Artificial neural network regression models in a panel setting: Predicting economic growth. *Economic Modelling, 91*, 148–154.

Jahromi, A. T., Stakhovych, S., & Ewing, M. (2016). Customer churn models: A comparison of probability and data mining approaches. *Looking forward, looking back: Drawing on the past to shape the future of marketing* (pp. 144–148). Cham: Springer.

Jerath, K., Fader, P. S., & Hardie, B. G. (2011). New perspectives on customer "death" using a generalization of the Pareto/NBD model. *Marketing Science, 30*(5), 866–880.

Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., & Abbasi, U. (2014). Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing, 24*, 994–1012.

Kingma, D.P., .& Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Klug, M., Barash, Y., Bechler, S., Resheff, Y. S., Tron, T., Ironi, A., et al. (2020). A gradient boosting machine learning model for predicting early mortality in the emergency department triage: Devising a nine-point triage score. *Journal of General Internal Medicine, 35*(1), 220–227.

Ładyżyński, P., Żbikowski, K., & Gawrysiak, P. (2019). Direct marketing campaigns in retail banking with the use of deep learning and random forests. *Expert Systems with Applications, 134*, 28–35.

Lemmens, A., & Gupta, S. (2020). Managing churn to maximize profits. *Marketing Science* forthcoming.

Martínez, A., Schmuck, C., Pereverzyev Jr., S., Pirker, C., & Haltmeier, M. (2020). A machine learning framework for customer purchase prediction in the non–contractual setting. *European Journal of Operational Research, 281*(3), 588–596.

McCarthy, D., Fader, P., & Hardie, B. (2016). V (CLV): Examining variance in models of customer lifetime value. *Available at SSRN 2739475*.

Medeiros, M. C., & Mendes, E. F. (2016). ℓ1-regularization of high-dimensional time-series models with non-Gaussian and heteroskedastic errors. *Journal of Econometrics, 191*(1), 255–271.

Mena, C. G., De Caigny, A., Coussement, K., De Bock, K. W., & Lessmann, S. (2019). Churn prediction with sequential data and deep neural networks. *A Comparative Analysis* arXiv preprint arXiv:1909.11114 .

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

Milošević, M., Živić, N., & Andjelković, I. (2017). Early churn prediction with personalized targeting in mobile social games. *Expert Systems with Applications, 83*, 326–332.

Nelson, S. D., Walsh, C. G., Olsen, C. A., McLaughlin, A. J., LeGrand, J. R., Schutz, N., et al. (2020). Demystifying artificial intelligence in pharmacy. *American Journal of Health-System Pharmacy, 77*(19), 1556–1570.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research, 12*, 2825–2830.

Platzer, M., & Reutterer, T. (2016). Ticking away the moments: Timing regularity helps to better predict customer activity. *Marketing Science, 35*(5), 779–799.

Qiao, J., Wang, L., & Yang, C. (2019). Adaptive lasso echo state network based on modified Bayesian information criterion for nonlinear system modeling. *Neural Computing and Applications, 31*(10), 6163–6177.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., et al. (2018). Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine, 1*(1), 1–10.

Reinartz, W. J., & Kumar, V. (2000). On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. *Journal of Marketing, 64*(4), 17–35.

Reutterer, T., Platzer, M., & Schröder, N. (2020). Leveraging purchase regularity for predicting customer behavior the easy way. *International Journal of Research in Marketing* in press.

Rudin, C., & Carlson, D. (2019). The secrets of machine learning: Ten things you wish you had known earlier to be more effective at data analysis. *Operations research & management science in the age of analytics* (pp. 44–72). INFORMS.

Salehinejad, H., & Rahnamayan, S. (2016). Customer shopping pattern prediction: A recurrent neural network approach. In *2016 IEEE symposium series on computational intelligence (SSCI)* (pp. 1–6). IEEE.

Schmittlein, D. C., Morrison, D. G., & Colombo, R. (1987). Counting your customers: Who-are they and what will they do next? *Management Science, 33*(1), 1–24.

Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2014). Model selection using database characteristics: Developing a classification tree for longitudinal incidence data. *Marketing Science, 33*(2), 188–205.
Schweidel, D. A., & Knox, G. (2013). Incorporating direct marketing activity into latent attrition models. *Marketing Science, 32*(3), 471–487.
Shi, R., Chen, H., & Sethi, S. P. (2019). A generalized count model on customers' purchases in O2O market. *International Journal of Production Economics, 215*, 121–130.
Smeekes, S., & Wijler, E. (2018). Macroeconomic forecasting using penalized regression methods. *International Journal of Forecasting, 34*(3), 408–430.
Soffer, S., Klang, E., Barash, Y., Grossman, E., & Zimlichman, E. (2020). Predicting in-hospital mortality at admission to the medical ward: A Big-Data machine learning model. *The American Journal of Medicine*.
Suh, E., Lim, S., Hwang, H., & Kim, S. (2004). A prediction model for the purchase probability of anonymous customers to support real time web marketing: A case study. *Expert Systems with Applications, 27*(2), 245–255.
Tamaddoni, A., Stakhovych, S., & Ewing, M. (2016). Comparing churn prediction techniques and assessing their performance: A contingent perspective. *Journal of Service Research, 19*(2), 123–141.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
Van Oest, R., & Knox, G. (2011). Extending the BG/NBD: A simple model of purchases and complaints. *International Journal of Research in Marketing, 28*(1), 30–37.
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.
Wang, J., Cai, Q., Chang, Q., & Zurada, J. M. (2017). Convergence analyses on sparse feedforward neural networks via group lasso regularization. *Information Sciences, 381*, 250–269.
Wang, J., Lai, X., Zhang, S., Wang, W. M., & Chen, J. (2020). Predicting customer absence for automobile 4S shops: A lifecycle perspective. *Engineering Applications of Artificial Intelligence, 89*, Article 103405.
Wood-Doughty, Z., Shpitser, I., & Dredze, M. (2020)Sensitivity analyses for incorporating machine learning predictions into causal estimates.
Xie, S.M. (2019). Neural network based parameter estimation method for the Pareto/NBD model. arXiv preprint arXiv:1911.01919.
Xie, S. M., & Huang, C. Y. (2020). Systematic comparisons of customer base prediction accuracy: Pareto/NBD versus neural network. *Asia Pacific Journal of Marketing and Logistics*.
Yu, R., An, X., Jin, B., Shi, J., Move, O. A., & Liu, Y. (2018). Particle classification optimization-based BP network for telecommunication customer churn prediction. *Neural Computing and Applications, 29*(3), 707–720.
Zhang, H.Y. (2008). Modeling discrete-time transactions using the BG/BB model.
Zhang, Y., Bradlow, E. T., & Small, D. S. (2015). Predicting customer value using clumpiness: From RFM to RFMC. *Marketing Science, 34*(2), 195–208.
Zhu, B., Baesens, B., & Van Den Broucke, S. K. (2017). An empirical comparison of techniques for the class imbalance problem in churn prediction. *Information Sciences, 408*, 84–99.
Zhu, J., Rosset, S., Tibshirani, R., & Hastie, T. J. (2003). 1-norm support vector machines. *Advances in Neural Information Processing Systems* (p. None).