http://dx.doi.org/10.15830/kjm.2021.36.3.133

Predicting Retail Consumers' Repeat Purchase Behaviors Using Machine Learning Methods

소비자들의 반복 구매 행동 예측을 위한 기계 학습 모델

Jun B. Kim • 김준범

ABSTRACT

Our goal in this paper is to predict retail consumers' repeat purchase behaviors using machine learning techniques and assess their predictive performance outcomes. To that end, we use individual-level, high-frequency transaction data from franchise retail stores. Our dataset is promising for machine learning applications given that consumer purchases in our empirical setting are more frequent and their patterns more complex than those used in consumer churn or adoption analyses in the past.

We report that the machine learning techniques of extreme gradient boosting and neural networks exhibit better performances overall than traditional models, but the extent of these improvements is marginal. Although underwhelming, our result is broadly aligned with earlier findings. We offer four conjectures behind this finding: suboptimal neural network design and training, linearly separable data, a suboptimal design with regard to feature engineering, and a need for more data. We further discuss the implications of each in the paper. Our paper will offer precursors and guidelines to academics and practitioners who may be interested in applying machine learning techniques to predict consumers' repeat purchase behaviors in retail settings.

Keywords: Neural Network (NN), Extreme Gradient Boosting (XGBoost), Repeat Purchase Prediction, Franchise Retail

This study was supported by the Institute of Management Research at Seoul National University.

Jun B. Kim | Associate Professor at Seoul National University(junbkim@snu.ac.kr)

pISSN 1229-456x, eISSN 2713-9913

초 록

본 논문은 기계학습 기술에 기반한 방법들을 사용하여 소비자들의 반복 구매 패턴을 예측하는 모델의 개발과 평가를 그 목적으로 한다. 소비자 이탈 및 소비자 획득 예측과 같은 기존 마케팅 응용에 사용된 데이터들에 비하여 프랜차이즈 소매 환경에서의 소비자 구매 패턴은 더 빈번하고 복잡하기 때문에 기계학습에 기반한 방법들을 적용할 때 그 예측율이 더 높을 것으로 기대한다.

프랜차이즈 소매 데이터 분석 결과, 기계 학습에 기반한 대표적인 방법인 엑스트림 그래디언트 부스팅(XGBoost) 및 신경망(neural network) 모델들이 전통적인 통계 방법에 비하여 그 예측력이 높지만, 그 예측력 개선의 정도가 크지 않다는 결론을 내린다. 이러한 결론은 기계학습을 이용한 방법을 비교 연구한 다른 분야들의 결과와 일치한다. 이러한 결과에 대하여 본 논문은 아래와 같은 네가지 가설을 제시하고 이에 대하여 논의한다. 제시하는 가설들은 최적화되지 않은 신경망 설계 및 훈련, 선형으로 분리 가능한 데이터의 특성, 비최적형태로의 데이터의 사용 및 더 많은 데이터의 필요성들이다. 특히 데이터에 관련된 사항이 가장 커다란 이유로 생각되며, 양질의 데이터의 획득과 더불어 최적의 feature engineering에 대한 노력이 더욱 필요한 것으로 보인다. 이 논문은 각종 소매 환경에서 인공 지능 기술을 사용하여 소비자 행동 예측을 수행하려는 마케팅 분야 학자들과 경영자들에게 실무적인 가이드를 제공한다.

핵심주제어: 신경망, Extreme Gradient Boosting (XGBoost), 소비자 반복 구매 예측, 프랜차이즈 소매

본 논문은 서울대학교 경영대학 경영연구소 연구비 지원으로 수행 되었습니다.

김 준 범 | 서울대학교 경영학과 부교수(junbkim@snu.ac.kr)

Ⅰ. Introduction

Our goal of this paper is to predict retail consumers' repeat purchase patterns using machine learning techniques and to assess their performance outcomes using individual-level and high-frequency data from a large franchise chain. Many industry observers and research firms forecast that the impact of Artificial Intelligence (AI) and big data will be most substantial in the marketing domain. For instance, McKinsey and Co. and the Harvard Business Review recently published a report that predicts that the marketing and sales function will be the top beneficiary of AI (Chui et al. 2018). Popular business media also report that more than 50% of big data applications may come from sales and marketing (Columbus, 2015). McKinsey and Co. also forecast that the retail sector will be among the top three beneficiaries from "predictive" models, behind only the travel and logistics sectors (Chui et al. 2018). Predictive models refer to a broad set of statistical or computational methods that are used to predict future events from the past. Businesses can use predictive models to detect early anomalies in the manufacturing process and avoid costly disruptions or breakdowns. In retail businesses, predictive models can serve as a better decision support system (DSS) and can help managers with their everyday decisions (Koehn et al. 2020). We see two benefits from a successful prediction model in our retail franchise setting. First, franchise store operators can use a prediction model for individual-level targeting by pushing out advertisements or coupons. Second, this model may lower costs by helping them make better product assortment and stocking decisions. We see the latter as a critical application area, as our franchise stores offer highly perishable products, and unsold items will hurt their profitability.

Reflecting the excitement and optimism across disciplines, a growing volume of research has applied machine learning techniques to various problems (김혜진, 이명구, 2021). Applications are found in medicine (e.g., Jeatrakul and Wong 2009), in credit default and credit rationing data (e.g., Sayeh and Bellier 2014; Adeodato et al. 2004), in consumer churn data (Ahn et al. 2019), and in consumer shopping data (e.g., Bakshi et al. 2018). These studies also compare new methods and traditional statistical models. Ahn et al. (2019) show that a neural network statistically outperforms logistic regression, although its marginal improvement is less than 2%. Adeodato et al. (2004) also report that the deep-learning model statistically outperforms logistic regression, but the degree of the improvement is again marginal at less than 1% in their case. Although their improvements are statistically significant, both papers report only minor performance improvements by the neural network.

There are review papers that surveyed past research that compared performance outcomes between neural networks and traditional statistical models. Dreiseitl and Ohno-Machado (2002) compiled a comprehensive survey of biomedical applications. In Table 2 of their paper, 69% of 61 papers report no statistical difference between a neural network and logistic regression. The rest of the surveyed papers report the outperformance of the neural network. In the end, they conclude that "there is no single algorithm that performs better than all other algorithms on any given dataset and application area."

More recently, Paliwal and Kumar (2009) compiled an even more comprehensive review, surveying articles across disciplines such as finance, health, manufacturing, and marketing. Their surveyed papers are very heterogeneous in terms of the domain, data, error measures, and validation methods. For instance, Limsombunchai et al. (2005) use consumer credit data and adopted a confusion matrix as the validation metric. They report that a neural network outperforms logistic regression with an accuracy rate of 86.62% vs 87.41%. However, the difference is quite small and does not accompany any statistical test. In Table 6 of their paper, Paliwal and Kumar (2009) offer a performance comparison between a neural network and traditional models across a number of surveyed articles. Conditional on binary classification tasks, 24 papers out of 37 report the outperformance of neural networks. For the remaining 13 papers, the neural network did not perform better. However, the surveyed papers again do not report any statistical test. Conditional on articles with statistical tests, 55% (17 out of 31) of the papers report the outperformance of a neural network. They conclude that neural networks outperformed in most of the cases or at least performed as well as other methods. There are two points that require further discussion with regard to the surveyed papers.

First, as Paliwal and Kumar (2009) point out, a very large fraction of the surveyed papers did not conduct statistical tests in their comparisons. This implies that once statistical tests are conducted, the performances of the neural network and logistic regression method may be similar. Second, the data sizes in the surveyed papers are all relatively small, ranging from a few hundred to a few thousand data points. The small datasets are understandable because these papers were prepared before the era of "big data."

The key differentiating aspect of this research is the data: we use individual-level, high-frequency transaction data from franchise retail stores. Using a dataset generated in a different empirical context, we aim to compare and test the performance capabilities of machine learning models and traditional models. Our dataset is promising for machine learning applications given that consumer purchases are more frequent and the corresponding patterns more complex. First, we observe a higher number of observations per consumer. Our data are generated in an urban setting in which consumers make frequent store visits due to the wide availability of our focal chain stores. Frequent store visits will translate into a higher number of observations per customer over time, which will be conducive to the data-hungry machine learning algorithms. In our data, we observe that consumers in the top percentile visit stores 1.06 times a week on average. In contrast, Guadagni and Little (1989) observe that the average number of weekly purchases by the 200 heaviest buyers in the coffee category is 0.32. In Siddarth et al. (1995), the weekly purchase rate for the "heaviest" buyer group in the laundry detergent category is far smaller at approximately 0.15. Second, consumer purchase patterns in our data are complex relative to those in earlier marketing applications. Recent applications of consumer data have mainly focused on one-time behavioral or status changes, such as life cycle status changes (e.g., Bakshi et al. 2018), consumer purchase conversions (e.g., Kohen et al. 2020), consumer churn (e.g., Ahn et al. 2019), or consumer credit default behaviors (e.g., Adeodato et al. 2004). In contrast, we use consumers' transaction panel data and predict repeated purchase behaviors by consumers. As a concrete example, the following can be considered. If we represent a consumer's "retained state" as 0 and "churned state" as 1, a typical customer may be represented with the string "000011111" in consumer churn data, indicating one permanent status change in the string. However, in our repeated purchase context, a consumer will be represented with a more complex string (e.g., "00010100001," in which 1 represents a purchase and 0 a non-purchase) with multiple changes over time. Multiple changes may be more challenging to predict than a single change, and machine learning may be suitable for such cases. Lastly, we apply XGBoost, a popular machine learning technique, in addition to a neural network to analyze our data.

The retail environment, by definition, is a big data industry (Dekimpe 2020). Our goal is to apply machine learning techniques to high-frequency franchise retailing data and compare the outcomes against those of traditional models. Our data and empirical setting are feasible for use with machine learning applications due to the high frequency and complexity of the data involved. With this paper, we hope to offer a practical guideline to marketing academics and practitioners who would be interested in applying machine learning techniques to a data-rich environment. The rest of the paper is organized as follows. In the next section, we introduce our data and describe our methods of feature preparation and model implementation. Next, we present the results and offer comparisons against baseline models. Finally, we discuss the implications of our results and conclude the paper.

Ⅱ. Application

1. Data

We provide a brief description of the data for our application. Our data come from a major franchise chain in Korea.[1] The franchise chain sells confectionery and beverage products to consumers and has a very comprehensive nationwide offline presence. Our dataset contains individual-level transaction records from multiple stores, all located in a major residential and business district in Seoul. In the data, we observe the complete transaction history of hundreds of thousands of consumers over a 15-month period starting from April of 2017. Among the observed data fields are the customer ID, timestamp, store ID, product purchased, and sales amount. For our optimization and prediction exercise, we randomly select 1,900 consumers who visited the stores at least 30 different days over 15 months. Such a selection process results in approximately 152,000 observations in our data. Given the number of fields and the long duration in the data, our data may be characterized as "narrow" and "tall;" we have a small number of columns but many rows.

2. Feature Preparation

Next, we discuss our data preparation method. First, we prepare and define dependent and independent variables for the development of our model. Independent variables (or "features" in deep-learning terminology) and dependent variables (or "labels" in deep-learning terminology) are defined as shown below. We start with our labels.

We set y_{it}=1 if consumer i $(=1,\cdots,I)$ makes a purchase on day $t(=1,\cdots,T)$ and set it to 0 otherwise. With this notation, we can represent consumer i's purchase history as $<y_{it}>$, $t=1,\cdots,T$. Then, our goal is to compute and predict the probability that consumer i makes a purchase on day t ($>s$) conditional on the past purchase history of $<y_{is}>$, or

$$Pr(y_{it}=1|<y_{is}>),\ s=t-1,...,1.$$

We use two different approaches to represent the sequence of $<y_{is}>$ in the feature representation. In our first approach, we adopt the RFM (recency, frequency, and monetary value) values. In RFM, consumers are concisely characterized as a vector of the recency, frequency, and monetary values of their past purchases. Researchers have recently used RFM metrics to characterize consumers in churn analyses (e.g., Mitrovic et al. 2017; Ahn et al, 2019). We follow their approach and use the following set of variables in our first

[1] To maintain confidentiality, the data provider wants to remain anonymous.

approach.

1. Recency: a pair (number of days between t and s, the day of the week of s), where s is one of the five most recent transaction dates
2. Frequency: purchase frequency for the last eight weeks up to t
3. Monetary Value: gross purchase amount for the last eight weeks up to t
4. Demographic variables of i such as age and gender

We use an 8-week time window for the operationalization of monetary value and frequency. We further operationalize the "recency" vector as follows. Assume that we are to predict the purchase incidence probability for today (t), that today is Monday, and that the last transaction (s) was made three days ago. Then, we encode the feature vector for today (t) as follows:

- we set $r_1=3$ as the number of days between t and s
- we set d_1 ="Friday" as the last purchase day in the feature.

We repeat this process for the five most recent transactions ($r_j, d_j, j=1,\cdots,5$) from the data and use them as the recency vector.

In our second approach of feature engineering, we use the full incidence vector of $< y_{is} >$,

$$< y_{is} > = < y_{it-1}, y_{it-2}, \cdots, y_{is-N-1}, y_{is-N} >,$$

in which each element of y_{is} takes a value of 1 if i makes a purchase on s, and 0 otherwise. In this implementation, we must set the value of N or the length of a moving time window for the incidence vector. In our application, we choose two values of 30 and 90. Two values for N will help us to investigate the value of longer panels in the prediction studies.

In summary, our first approach for feature engineering is an application of RFM values with minor variations. That is, we include five most recent transaction days in our operationalization of the "recency" variable. Our second approach for feature engineering is based on the full incidence history, which is a direct representation of the data. It also contains more granular information compared to the first approach. However, a limitation of the full incidence vector approach is that it is unidimensional given that it does not utilize sales amount data. In contrast, RFM variables utilize these data through the monetary value variable. Therefore, it is not clear which approach will lead to a better performance. Next, we briefly discuss the two machine learning techniques used in our prediction exercise. Our choice of two models is based on a recent paper that reported the superior performance of extreme gradient boosting and a neural network compared to other methods (Orzechowski et al. 2018).

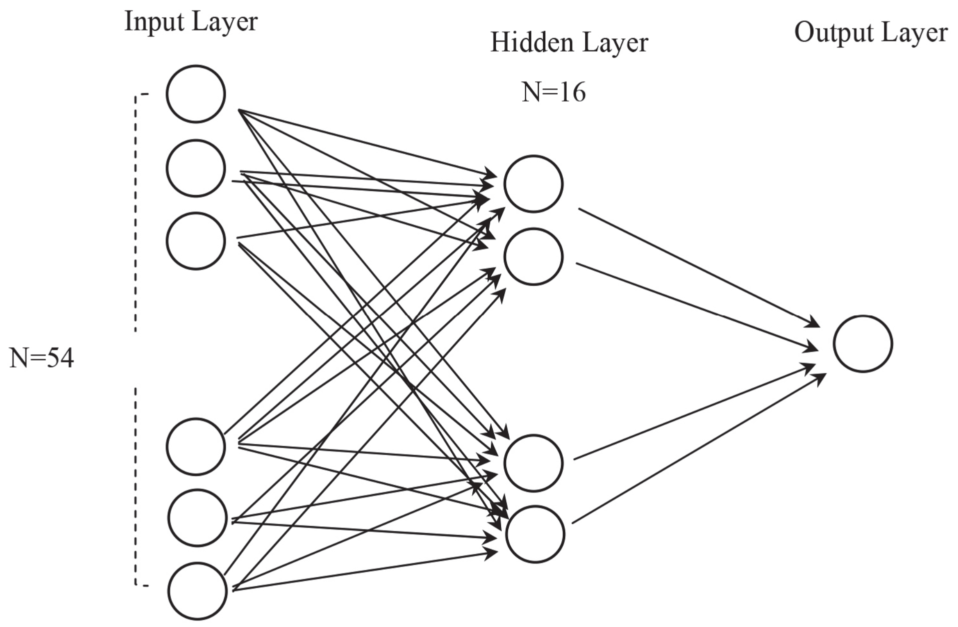
3. Neural Network

Inspired by related concepts in neural biology, the neural network has become a popular tool for classification tasks across disciplines. One key differentiating aspect of recent neural networks when compared to the "old" rule-based AI approach is the relationship between rules and data. In deep-learning techniques in the field of modern AI, no rules are explicitly encoded beforehand. Instead, known answers and "training data" are used as inputs to the neural network, and the outputs are the "rules." These "data-driven" rules are then used to obtain answers from other data. From a pure modeling perspective, the key advantage of a neural network is that it can approximate any nonlinear function with a high degree of accuracy (Leshno et al. 1993).

To operate a neural network, one must decide on its architecture by setting various hyper and tuning parameters. Figure 1 shows an instance of a feed-forward neural network model that we implemented with RFM values as the input vector. In general, the numbers of layers and neurons per layer determine the complexity of the neural network, and more complex models can capture higher-order patterns in the data. For our neural network, we use one hidden layer because such a model can approximate any arbitrary continuous function with sufficient accuracy (Zhong et al. 2017). Consequently, a single hidden layer model will suffice for a vast majority of binary classification tasks. Although there is no explicit formula for the optimal number of neurons per layer, the general rule of thumb is that it must lie between the numbers of inputs and output neurons. The performance of neural networks also depends on other hyper-parameters, such as the learning rate and batch size. The learning rate determines the speed at which the optimizer moves towards the optimal weights during the optimization process, with a high value meaning that the optimizer will take large steps when updating the weights in the neural network. The batch size is the number of training samples fed into the training model when updating the weights in one iteration or epoch. A small batch size means that the training samples will be broken into smaller sets, each of which will then be used during the weight update process. We try different combinations of neurons, step sizes, penalty values, and batch sizes to ascertain the best performing model during our optimization process. Table 1 shows the different values of hyper and tuning parameters we used during our optimization sessions.

〈Figure 1〉

An instance of feed-forward neural network architecture with one input, one hidden, and output layer with RFM features. There are one input layer with 54 inputs, one hidden layer with 32 neurons, and one output layer. The length of input vector (N=54) is determined by multiple factors such as gender (M/F), age group (20,30,40,50, Unknown), and RFM variables for a customer. RFM variables reflect monetary value (M), frequency (F) and a recency vector (R) corresponding to the number of lapsed days and day of the week (Monday to Sunday) from five latest transactions. Note that all discrete variables such as gender, age group, and day of the week are encoded as dummy variables in the vector.



〈Table 1〉 Hyper and tuning parameter values used for Neural Network optimization

Hyper parameters	Values
Learning rate	{0.001, 0.003}
Number of neurons in the hidden layer	{8, 32}
Batch size	{40, 160}
Penalty	{1e-4, 1e-5}

4. Extreme Gradient Boosting

Gradient boosting is an effective training algorithm for ensemble models that use boosting (Kelleher et al. 2020). Often characterized as the "wisdom of experts" approach, gradient boosting constructs and combines a set of "weak" models into a "strong" one. That is, the algorithm adds new, lower level "trees" that sequentially fit the residuals from existing, higher level "trees." Gradient boosting repeats this sequential, top-down, tree-adding process until specific criteria are met. XGBoost (extreme gradient boosting) is a special version of the gradient boosting model with more regularization. Since its inception in 2014, XGBoost has been a popular choice among participants in Kaggle, a well-known online data science competition community.[2] Given its popularity and recent success, we choose extreme gradient boosting as a machine learning technique to predict consumers' repeat purchase patterns in our data.

The performance of extreme gradient boosting also depends on the set of tuning and hyper-parameters used. One such parameter is *max_depth*, the maximum number of "tree layers" allowed from the root to the farthest node (leaf). A higher parameter value means a deeper and more complex tree, which may lead to overfitting with regard to the training data. Another important tuning parameter is the learning rate, a correction factor for new trees when they are added to the model. A higher learning rate means that corrections by new trees will be fully reflected to minimize the residual error during the optimization process. All of the parameters described above play a crucial role in the model's performance trade-off between training and validation fit. Extreme gradient boosting tends to learn very fast, and it is essential to monitor the optimization process and avoid model overfitting. Readers can refer to Table 2 for a list of the parameters and their values used during the tuning process.

As a benchmark for comparison, we use logistic regression. Logistic regression is the most popular model for binary classification tasks, with applications found across disciplines. Because logistic regression often served as the baseline model in earlier works, we also use it here.

〈Table 2〉 Hyper and tuning parameter values used for Extreme Gradient Boosting learning process

Parameters	Values
Learning rate	{0.04, 0.2}
Max depth	{9, 10}
Minimum child weights	{3, 4}
Gamma	{1, 4}
Subsample	{0.8, 1.0}
N_estimator	{100, 500}

5. Model Optimization

We use various open-source packages for model implementation and optimization. We use a logistic regression package from Sklearn library,[3] a neural network package from Keras' Sequential library,[4] and an extreme gradient boosting package

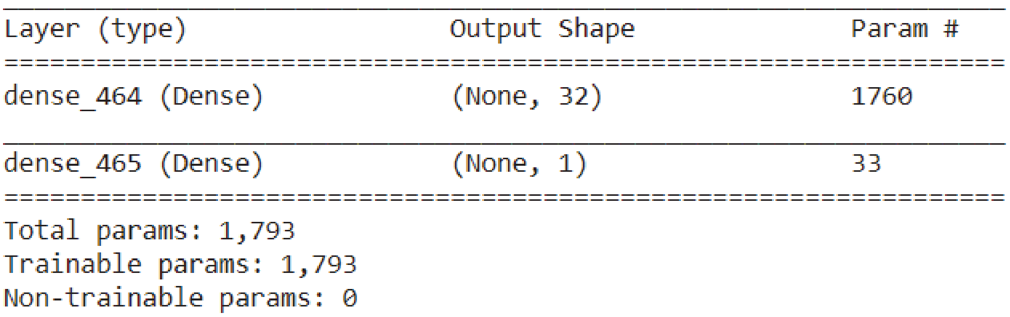
[2] https://en.wikipedia.org/wiki/XGBoost

[3] https://scikit-learn.org/stable/

[4] https://keras.io/guides/sequential_model/

〈Figure 2〉

Numbers of weights to optimize in the hidden layer (dense_6) and output layer (dense_7) in a feed-forward neural network. The neural network has 32 inputs to the hidden layer and 16 neurons in the hidden layer. The output payer has 32 inputs and 1 output. The total number of weights to optimize is 1,793.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_464 (Dense)            (None, 32)                1760
_________________________________________________________________
dense_465 (Dense)            (None, 1)                 33
=================================================================
Total params: 1,793
Trainable params: 1,793
Non-trainable params: 0
```

from XGBoost library.[5] Because training with XGBoost and a neural network requires an extensive tuning process during the optimization process, we use an approach similar to a grid search. That is, we attempt different combinations of hyper and parameter tuning (see Tables 1 and 2), monitor the optimization process, and avoid overfitting. Figure 3 shows an example of a training session trajectory of a neural network. In the figure, both the accuracy and AUC increase as optimization progresses, and both are higher with the training set than with the validation set. We repeat this process across different combinations of tuning and hyper parameters. We adopt ten-fold cross-validation for model selection. Cross-validation is often adopted when the dataset is small or when one must study the statistical properties of a model. In the ten-fold cross-validation scheme, we divide all of the data into ten subsets. We then use nine subsets as the optimization data and the remaining one subset as the test data. During each optimization, we use 75% of the optimization data as the training set and the remaining 25% as the validation set. Therefore, in each optimization step, we use 67.5% (=90% *75%) of the data as the training set, 22.5% (=90%*25%) as the validation set, and 10% as the test set. We repeat this process ten times with different combinations of sets during the ten-fold cross-validation process. We use the mean and standard deviations across ten sessions for statistical testing and for model selection.

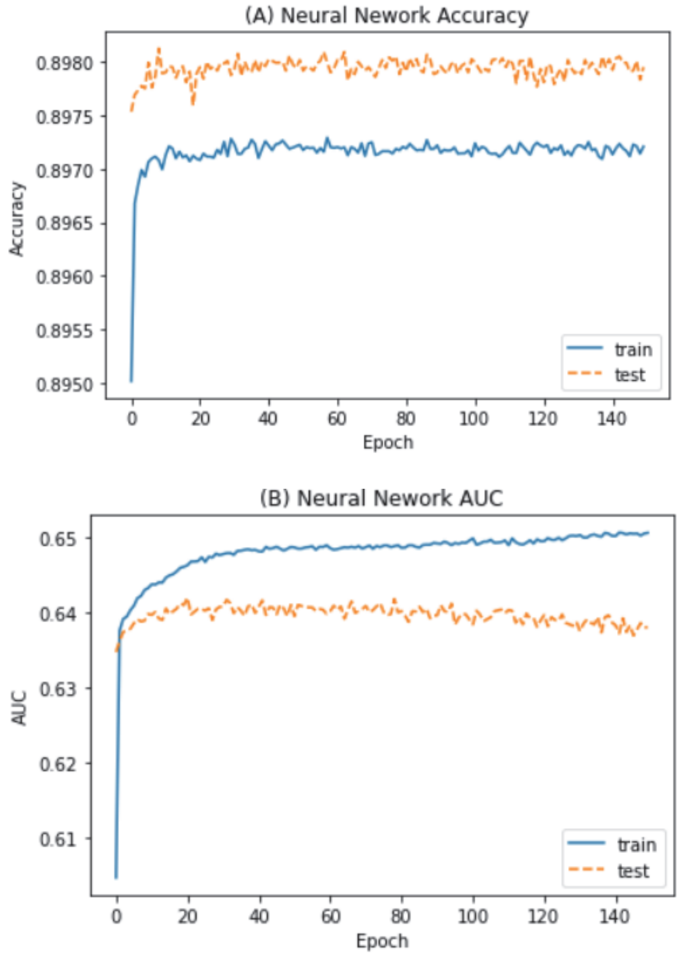
Ⅲ. Results

We present our results using the two metrics of the accuracy and the area under the curve (AUC) for model comparison. In binary classification task, accuracy is defined as the ratio of true predictions compared to actual observations. Although accuracy is a popular metric for assessing the performance of statistical models, it is less ideal for unbalanced data. For intuition, we consider a dataset in which 90% of all labels are overwhelmingly “negative.” Then, one can achieve 90% accuracy by making a simple prediction that all are “negative.” In unbalanced data, the area under the curve (AUC) may be a better option. AUC computes the probability that a model correctly ranks a random point in a positive area higher than a random point in a negative area. AUC discriminates between correct and

[5] https://xgboost.readthedocs.io/en/latest/

〈Figure 3〉

Accuracy and AUC charts for one training session in a neural network. Both accuracy and AUC for the training data set are higher than those of test data.



incorrect predictions and is therefore favored between the two metrics (Ling et al., 2003). For completeness, we use both metrics in this subsection.

At this point, we present our results. Table 3 shows the accuracy values across different datasets (training, validation, and testing) and different methods with RFM features. We note that the accuracy values are quite similar across the datasets and methods, all being between 0.877 and 0.880. Table 4 shows the AUC outcomes across datasets and methods. First, the AUC values are higher with the training dataset compared to the other two datasets across all models. For the training dataset, extreme gradient boosting shows the highest AUC compared to the other two methods. For the test sets, AUC from extreme gradient boosting (0.640) is higher than both logistic regression (p-value = 0.005, t-value = -2.9) and the neural network (p-value = 0.002, t-value = -3.3). Specifically, extreme gradient boosting outperforms logistic regression by 5%, and our improvement margin is higher than those reported in earlier works on consumer churn analysis (e.g., Ahn et al. 2019). Overall, our result shows that extreme gradient boosting exhibits the best in-sample and out-of-sample performances with RFM variables as features.

Table 5 shows the model performances using full incidence vectors as features. Going through and comparing the values in Tables 4 and 5, we observe that the AUC values in Table 5 are in general higher than those in Table 4. This means that the full incidence vector as a feature appears to be a better choice for our prediction task. Column (1) in Table 5 shows the AUC values of the training set with 30-day lookback across different methods. We note that extreme gradient boosting exhibits the highest AUC compared to logistic regression (p-value = 0.00, t-value = -4.76) and the neural network (p-value = 0.01, t-value = -2.5). In column (2), the AUC values are higher with a longer lookback period for both extreme gradient boosting and the neural network.

Next, we compare the AUC values with the test data in columns (5) and (6). First, in a comparison between *N*=30 and *N*=90, we find that the AUC values are in general higher for *N*=90 than for *N*=30. For instance, between columns (5) and (6) for logistic regression, AUC improves from 0.642 (*N*=30) to 0.648 (*N*=90), although the difference is not statistically significant. For the neural network case, the AUC value improves from 0.646 to 0.653. In column (6)

〈Table 3〉 Accuracy values for three methods with RFM features. Standard deviations are in the parenthesis

Method	Train	Validation	Test
Logistic Regression	0.877 (0.005)	0.878 (0.004)	0.877 (0.040)
XGBoosting	0.880 (0.004)	0.879 (0.004)	0.878 (0.039)
Neural Network	0.877 (0.004)	0.878 (0.004)	0.877 (0.040)

〈Table 4〉 Area Under Curve (AUC) for three approaches with RFM features. Standard deviations are in the parenthesis

Method	Train	Validation	Test
Logistic Regression	0.639 (0.009)	0.643 (0.009)	0.610 (0.022)
XGBoosting	0.715 (0.017)	0.669 (0.010)	0.640 (0.024)
Neural Network	0.637 (0.009)	0.640 (0.009)	0.606 (0.022)

〈Table 5〉 Area Under Curve (AUC) for three different approaches with past incidence vector of 30- and 60- day lookback windows. Standard deviations are in the parenthesis

Method	Train		Validation		Test	
	(1)	(2)	(3)	(4)	(5)	(6)
	N=30	N=90	N=30	N=90	N=30	N=90
Logistic Regression	0.667 (0.008)	0.648 (0.009)	0.662 (0.009)	0.648 (0.008)	0.642 (0.024)	0.648 (0.028)
XGBoosting	0.684 (0.008)	0.707 (0.014)	0.667 (0.009)	0.674 (0.008)	0.645 (0.027)	0.651 (0.029)
Neural Network	0.675 (0.009)	0.683 (0.009)	0.666 (0.009)	0.677 (0.008)	0.646 (0.027)	0.653 (0.030)

with N=90, the nominal values of the AUC for both gradient boosting and the neural network are higher than that of logistic regression, although neither outcome is statistically significant. Lastly, in column (4) using the validation set and N=90, the neural network exhibits a higher AUC value compared to logistic regression (p-value= 0.00, t-stat = -8.1).

We conclude the following from our results. First, from the perspective of feature preparation, the full incidence vector is a better choice than RFM in our repeat purchase prediction exercise. This implies that feature engineering is an important topic for prediction models of repeat purchase. Second, extreme gradient boosting and the neural network overall exhibit better prediction performance with test sets than logistic regression with the RFM feature. Third, both the neural network and extreme gradient boosting show better prediction performance outcomes against logistic regression with the full incidence vector as a feature, although the improvement is not always statistically significant. Last, XGBoost appears to be better choice than a neural network, the same conclusion reached by Orzechowski et al. (2018). We conclude that machine learning methods are a dominating choice when tasked with modeling repeated consumer purchase predictions.

Our finding overall is well aligned with findings in the literature reporting that neural networks in general exhibit marginally better performance than traditional models. A natural question that arises is related to why machine learning methods such as neural networks, which can approximate any nonlinear function, do not exhibit far superior performance than a baseline model. We suggest a few conjectures as potential answers to this question and discuss their implications with regard to academics and practitioners. First, given the complexity in the architecture and design of a machine learning methods, it may be that we failed to find the optimal configuration, leading to sub-optimal architecture and design outcomes in our exercise. However, this conjecture may be less appealing, as we adopted a strategy similar to a grid search and attempted to identify the best architecture and design. We also attempted multiple hidden layers in the design of the neural network, only to find similar performance outcomes. Nonetheless, it remains possible that the use of more complex neural network models such as a recursive neural network (RNN), if applicable, may lead to better performance.[6]

A second probable explanation lies in the nature of the data in our empirical setting; our data may be linearly separable. Two subsets of A and B are linearly separable if there exists a hyperplane that completely separates points in the two sets of A and B (Elizondo 2006). If the data are linearly separable, one may not need a complex, nonlinear classifier to separate points in two subsets. Therefore, if our data are indeed linearly separable, logistic regression may suffice for a binary classification task. To the best of our knowledge, there does not seem to be a formal test for the linear separability of data;[7] hence, as leave this as an open possibility.

The third probable reason is in the way we use the data for feature engineering, and in relation to this our approach here may be suboptimal. There are alternative means by which to construct a set of independent variables or “features” from

6. We faced the following challenges when applying a RNN to our dataset. First, although a RNN is often applied to time series data, we have panel data here, meaning that it is cross-sectional and longitudinal at the same time. Second, the initiations of purchases across consumers are all different. One option is to apply a RNN to each consumer. Due to these challenges, we adopted a FFN with long time windows.

7. Several online sources mention a couple of approaches, such as a specific application of a SVM (support vector machine) as an informal way to diagnose the linear separability of data.

our transaction data. In this paper, we adopted two different approaches and showed that the full incidence vector appears to be a better choice than RFM variables. However, there may be multiple ways to construct the features as inputs to machine learning models. Lastly, it is also possible that we lack quality data in our exercise and that additional data, if any exist, may lead to a more accurate prediction (송인성, 2020). Note that although our data are generated at a high frequency (i.e., many rows), they are narrow (i.e., a small number of columns); therefore, we may be missing important variables such as environmental factors (e.g., weather information and local events) that may have critically affected the predictions of consumer purchase behaviors. If available, these data may improve the performance of the proposed model.

The discussion above has implications that may be useful to other academics and to practitioners who may be interested in or who may have plans to develop predictive models in their own retail empirical settings. First, our discussion implies that it is very important to secure high-quality data for machine learning applications. For our application, we only utilized an internal dataset, which led to a data structure that was "long but "narrow." However, researchers must strive to acquire other relevant data, even if they are external. The use of additional datasets in machine learning models may lead to a better predictive model.

Second, for a given dataset, it may be important to identify the best feature sets. In our application, we assessed two different feature engineering approaches and found that the full incidence matrix outperforms RFM variables. Theoretically, we can test different combinations of features and identify the best performing set.

In any big data project, it is well documented that the front end of the project, or data-related tasks such as cleansing, preparation, and operationalization, takes much of the AI project time (Economist, 2020). Our findings and discussions also imply that the front end of the project not only takes a disproportionate amount of time but that it also plays a critical role in the overall success of a machine learning project. This means that in the era of big data, researchers must understand fully the strengths and limitations of their data before committing themselves to analyses. Given that our data and empirical setting are typical to those in other online and offline retailers, our findings and discussion may serve as a pragmatic guideline for other academics and practitioners who may be interested in developing predictive models in their own empirical settings.

Ⅳ. Conclusion

In this paper, we conduct a prediction exercise, focusing on consumers' repeat purchase behaviors using machine learning techniques. To that end, we use individual-level, high-frequency consumer panel data from a large franchise retailer. Our results show that extreme gradient boosting and a neural network outperform logistic regression overall in terms of the AUC. Nonetheless, the extent of the performance improvement is marginal and not always statistically significant. Although underwhelming, our result appears to be well aligned with earlier findings in other disciplines; previous studies also conclude that there is no single algorithm that outperforms the rest on a given dataset and application. We discuss a few probable reasons for our result, specifically why machine learning methods, which can approximate any nonlinear function, do not exhibit far superior performance than a benchmark model. We suggest four probable explanations - suboptimal design and training of the proposed neural network, linearly separable data, sub-optimal use of data in the feature preparation step, and the need for more quality data. We provide an in-depth discussion of each item and leave further investigations to future research.

〈Received April 14. 2021〉
〈1st Revised June 27. 2021〉
〈Accepted July 12. 2021〉

References

김혜진, 이명구(2021), "마케팅 분야의 머신러닝 연구 동향 분석," *마케팅연구,* 36(1), 1-25.

송인성 (2020), "마케팅 애널리틱스에서 머신 러닝 기법 활용의 한계," *경영논집,* 54, 39-57.

Ahn, Yongil, Dongyeon Kim, and Dong-Joo Lee (2019), "Customer Attrition Analysis in the Securities Industry: A Large-scale Field Study in Korea," *International Journal of Bank Marketing*, 38(3), 561-577.

Adeodato, Paulo J., Germano C. Vasconcelos, Adrian L., Arnaud, Roberto A. Santos, Rodrigo C. Cunha, and Domingos S. Monteiro (2004), "Neural Networks vs Logistic Regression: A Comparative Study on a Large Data Set," *International Conference on Pattern Recognition,* Cambridge UK, 355-358.

Bakshi, Nikhil A., Prithviraj R. Kolan, Bibek Behera, Naveen Kaushik, and Ansari M. Ismail (2018), "Predicting Pregnant Shoppers Based on Purchase History Using Deep Convolutional Neural Networks," *Journal of Advances in Information Technology*, 9(4), 110-116.

Chui, Michael, Nicolaus Henke, and Mehdi Miremadi (2018), "Most of AI's Business Uses Will Be in Two Areas," *Harvard Business Review*, (July 20), 1.

Columbus, Louis (2016), "Ten Ways Big Data Is Revolutionizing Marketing and Sales," (May 9), *Forbes*. https://www.forbes.com/sites/louiscolumbus/2016/05/09/ten-ways-big-data-is-revolutionizing-marketing-and-sales/?sh=4fe9ec6921cf.

Dekimpe, Marnik G. (2020), "Retailing and Retailing Research in the Age of Big Data Analytics," *International Journal of Research in Marketing*, 37(1), 3-14.

Dreiseitl, Stephan, and Lucila Ohno-Machado (2002), "Logistic Regression and Artificial Neural Network Classification Models: A Methodology Review," *Journal of Biomedical Informatics,* 35(5-6), 352-359.

The Economist (2020), "For AI, Data Are Harder To Come By Than You Think," (June 13), https://www.economist.com/technology-quarterly/2020/06/11/for-ai-data-are-harder-to-come-by-than-you-think

Elizondo, David (2006), "The Linear Separability Problem: Some Testing Methods," *IEEE Transactions on Neural Networks*, 17(2), 330-344.

Guadagni, Peter M., and John D. Little (1983), "A Logit Model of Brand Choice Calibrated on Scanner Data," *Marketing Science,* 2(3), 203-238.

Jeatrakul, P. and K. W. Wong (2009), "Comparing the Performance of Different Neural Networks for Binary Classification Problems," *Eighth International Symposium on Natural Language Processing*, 111-115.

Kelleher, John D., Brian M. Namee, and Aoife D'Arcy (2020), *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*, Second Edition, MIT Press.

Koehn, Dennis, Stefan Lessmann, and Markys Schaal (2020), "Predicting Online Shopping Behaviour from Clickstream Data Using Deep Learning," *Expert Systems with Applications*, 150, 113342.

Leshno, Moshe, Vladimir Y. Lin, Allan Pinkus, and Shimon Schocken (1993), "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural Networks*, 6(6), 861-867.

Ling, Charles X., Jin Huang, and Harry Zhang (2003), "AUC: A Better Measure Than Accuracy in Comparing Learning

Algorithms,” *Conference of the Canadian Society for Computational Studies of Intelligence*, 329-341.

McKinsey Global Institute (2018), “Notes from the AI Frontier Insights from Hundreds of Use Cases,” McKinsey & Company.

Mitrovic, Sandra, Gaurav Singh, Bart Baesens, Wilfried Lemahieu, and Jochen De Weerdt (2017), “Scalable RFM-enriched Representation Learning for Churn Prediction,” *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 79-88.

Orzechowski, Patryk, William La Cava, and Jason H. Moore (2018), “Where Are We Now? A Large Benchmark Study of Recent Symbolic Regression Methods,” *Proceedings of the Genetic and Evolutionary Computation Conference*, 1183-1190.

Paliwal, Mukta, and Usha A. Kumar (2009), “Neural Networks and Statistical Techniques: A Review of Applications,” *Expert Systems with Applications*, 36(1), 2-17.

Sayeh, Wafa, and Annie Bellier (2014), “Neural Networks versus Logistic Regression: The Best Accuracy in Predicting Credit Rationing Decision,” In *World Finance & Banking Symposium,* 1-22.

Siddarth, Sivaramakrishnan, Randolph E. Bucklin, and Donald G. Morrison (1995), “Making the Cut: Modeling and Analyzing Choice Set Restriction in Scanner Panel Data,” *Journal of Marketing Research*, 32(3), 255-266.

Zhong, Kai, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon (2017), “Recovery Guarantees for One-hidden-layer Neural Networks,” *International Conference on Machine Learning,* 4140-4149.