Profit-Driven Weighted Classifier with Interpretable Ability for Customer Churn Prediction

Ping Jiang[1], Zhenkun Liu[2]*, Mohammad Zoynul Abedin[3], Jianzhou Wang[4], Wendong Yang[5], Qingli Dong[6]

[1]School of Statistics, Dongbei University of Finance and Economics, Jianshan Road, Shahekou District, Dalian, Liaoning Province 116012, China

[2]School of Management, Nanjing University of Posts and Telecommunications, No 66 Xinmofan Road, Gulou District, Nanjing, Jiangsu Province 210023, China

[3]Department of Accounting and Finance, School of Management, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, Wales, the United Kingdom

[4]Institute of Systems Engineering, Macau University of Science and Technology, Taipa Street, Macao 999078, China

[5]School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, Shandong Province 250014, China

[6]School of Economics and Management, Dalian University of Technology, No2 Linggong Road, Ganjingzi District, Dalian, Liaoning Province 116081, China

* Corresponding author: Zhenkun Liu, Tel.: +86 18205084738

E-mail addresses: Ping Jiang (pjiang@dufe.edu.cn);

Zhenkun Liu (liuzk@njupt.edu.cn);

Mohammad Zoynul Abedin (m.z.abedin@swansea.ac.uk);

Jianzhou Wang (wangjz@dufe.edu.cn);

Wendong Yang (hshwendong@hotmail.com);

Qingli Dong (qinglidong@dlut.edu.cn)

Funding

This research was supported by the National Social Science Foundation of China (Grant No. 21CTJ011).

Financial interests and non-financial interests

The authors have no relevant financial or non-financial interests to disclose.

Profit-Driven Weighted Classifier with Interpretable Ability for Customer Churn Prediction

Abstract

Customer churn prediction methods aim to identify customers with the highest probability of attrition, improve the effectiveness of customer retention campaigns, and maximize profits. However, previous studies have relied on a single classifier, leading to suboptimal predictive results. To address this issue, we propose a novel profit-driven weighted classifier that integrates a weighted strategy with multiple profit-driven ensemble members. We employ an artificial hummingbird optimization algorithm to determine the optimal weight coefficients of the profit-driven ensemble members based on the expected maximum profit criterion. We then calculate the Shapley additive explanation value to further improve the interpretability of the proposed weighted classifier. We conducted experiments and statistical tests on eight real-world datasets from different industries. The results show that the proposed weighted classifier significantly improves profits compared with comparative classifiers and provides strong interpretability based on the Shapley additive explanation value.

Keywords: Expected maximum profit criterion; Profit-driven weighted classifier; Profit maximization; Shapley additive explanation value

Profit-Driven Weighted Classifier with Interpretable Ability for Customer Churn Prediction

1. Introduction

Currently, business analysts are focusing their research efforts on predicting various aspects of customer behavior, including customer credit, churn, response, and purchase prediction [1–3]. Customer churn refers to the termination of a relationship between a client and an enterprise, which inevitably reduces revenues and increases the cost of acquiring new customers. To remain competitive, companies are shifting their business strategies from selling products and services to developing advanced customer-centric information systems and conducting customer relationship management to prevent customer churn [4–6]. Previous research has shown that it costs five to six times more to acquire new customers than to retain existing ones [6,7]. In contrast, loyal and stable customers are less susceptible to competitive marketing efforts and generate higher profits. As a result, enterprises conduct customer retention campaigns to identify clients who are likely to defect to competitors and offer incentives to keep them from leaving. However, identifying individuals with a tendency to churn from large customer groups remains a challenge. To maximize profits and maintain a good reputation, companies need to explore high-quality classifiers that can successfully identify churners.

Customer churn prediction (CCP) is a binary classification problem that aims to infer customer churn behavior based on customer information using classifiers [8]. Companies currently possess a wealth of customer behavioral and historical information that can be leveraged to create effective predictors using machine learning technologies. Commonly used machine learning technologies in CCP include artificial neural networks (ANNs) [9–11], support vector machines (SVMs) [8], decision trees [5,12], and ensemble techniques [13–15]. However, studies have shown that individual technologies are not always effective due to the modeling complexity of CCP [16].

This study proposes a profit-driven CCP classifier that integrates a weighted strategy, an artificial hummingbird optimization algorithm (AHA), multiple profit-driven ensemble members, and Shapley additive explanations (SHAP) to compensate for the limitations of the current studies. To the best of our knowledge, Lessmann et al. (2021) proposed the most recently weighted model for customer churn prediction [17]. Compared to Lessmann et al. (2021), the proposed weighted classifier (PWC) has stronger profitability and interpretation. The detailed improvements in PWC and the contributions of our study are as follows:

First, the AHA is employed to globally optimize the weight coefficients based on the predictions of multiple profit-driven ensemble members with the goal of profit maximization. In a previous study [17], the hill-climbing method was used to search for local optimal solutions to determine the weights of the ensemble members. In our study, the AHA considers all ensemble members simultaneously to determine the globally optimal complementary combination of ensemble members. Experimental results show that the PWC is more profitable than profit-driven individual classifiers and comparative classifiers realized by many excellent ensemble strategies, such as stacking, model selection, average weighted, rank-based weighted, performance-based weighted, hill climbing-based weighted, statistical indicator-driven weighted, and genetic algorithm-based weighted strategies.

Second, the weighted SHAP values combine the weighted classifier and the SHAP values of the ensemble members, which provides interpretability for the weighted classifiers. This is because the essence of the weighted classifier is to weigh the predictions of multiple ensemble members, and the SHAP value exhibits local accuracy and consistency [18]. By calculating the weighted SHAP value, PWC has high explanatory power and can provide local and global explanations for decision-makers. However, the limitations discussed in existing research [17] claim that the newly developed weighted classifier is a black model without interpretation ability. Therefore, compared with the newly developed weighted classifier, the PWC improves the profitability of customer churn prediction and uses the weighted SHAP value to obtain explanatory power.

With PWC, marketing managers can take more aggressive actions with relatively higher accuracy. Managers can provide personalized services to clients who are likely to leave the company, which can increase customer satisfaction. This has direct managerial applications, as new customer acquisition is expensive in a highly competitive and constantly changing environment. In addition, decision-makers can use the knowledge gained from PWC to improve decision-making processes, such as targeted marketing, from a profit-oriented perspective.

Section 2 presents the related literature on profit-oriented CCP, and ensemble classifiers in CCP. **Section 3** introduces evaluation metrics and the proposed weighted classifier. **Section 4** presents the experimental design, and **Section 5** discusses the simulation results. Finally, **Section 6** presents the conclusions and discussion.

2. Literature review

This section discusses the literature on profit-driven CCP, and ensemble classifiers for CCP.

2.1. Research on profit-driven customer churn prediction

CCP is a constantly evolving task due to changes in technology and competition, which lead to changes in retention activities and the impact of retention campaigns on clients over time [19]. Therefore, it is imperative to establish a reliable framework for CCP to improve the identification abilities of churners in an ever-changing environment.

If a customer switches to a competitor, the current sector loses customer lifetime value (CLV). If the customer is retained through a retention campaign, the company gains CLV but loses retention costs. Otherwise, the firm incurs both CLV and retention costs [20]. There were significant differences between misclassifying non-churners as churners and misclassifying churners as non-churners. Therefore, it is recommended that the benefits of correct classification and the costs of misclassification be distinguished and considered simultaneously to maximize the profits of retention campaigns [21].

To estimate the profits of retention campaigns, Verbeke et al. (2012) provided a profit-related metric, the maximum profit criterion (MPC), which considers both the costs of a retention campaign and the benefits of retaining customers [20]. Verbraken et al. (2013) proposed an extension of the MPC, the expected maximum profit criterion (EMPC), which introduces an uncertainty measure for the probability of recovering potential churners [22]. The central function of MPC and EMPC is to evaluate multiple predictors in business scenarios [2,23,24]. Traditional indicators, including accuracy, precision, and AUC, are suboptimal for evaluating classifiers from a business perspective [22]. In addition, researchers have incorporated profit-related metrics into the model construction process. For example, Maldonado et al. (2015) proposed a framework that added profit-related metrics to feature selection and model construction based on SVMs [25]. Stripling et al. (2018) proposed ProfLogit, which uses a genetic algorithm in the training process to maximize EMPC [26]. Similarly, Höppner et al. (2018) incorporated a profit metric to optimize tree construction using an evolutionary algorithm to maximize EMPC [12]. Inspired by the above studies, Maldonado et al. (2020) presented three methods for profit-driven CCP [27]. The experimental results verify the effectiveness of employing a profit measure to evaluate and calibrate the classifiers. **Table 1** lists other relevant studies on profit-oriented churn prediction.

Table 1
Relevant studies on profit-oriented CCP

Reference	Datasets Field	Datasets Number	Research Highlights	Weighted Classifiers	Classifier Selection Form	Interpretable Method	Statistical Evaluation
Verbeke et al. (2012) [20]	Telecom	11	This study proposes a profit-centric performance indicator that is more effective in generating profits than traditional statistical metrics.	-	-	-	Friedman test and Nemenyi test
Verbraken et al. (2013) [22]	Telecom	10	It proposes the EMPC based on a cost-benefit analysis framework.	-	-	-	-
Verbraken et al. (2014) [24]	Telecom	5	It investigates the predictive ability of several Bayesian Network models for churn prediction.	-	-	-	Friedman test and Nemenyi test
Maldonado et al. (2015) [25]	Telecom	3	It presents a profit-driven method for classifier construction and variable selection.	-	-	-	-
Zhang et al. (2016) [28]	Telecom	1	It presents a profit function to maximize profit in different situations.	-	-	-	-
Stripling et al. (2018) [26]	Telecom	9	It proposes ProfLogit, which maximizes the EMPC by using genetic algorithm in the training process.	-	-	-	-
Lee et al. (2018) [29]	Online game service	1	This study presents a classifier that considers the expected profit of the online game.	-	-	-	-
Óskarsdóttir et al. (2018) [23]	Bank and Telecom	7	This study extends the EMPC by considering the variability of customer lifetime values.	-	-	-	-
Höppner et al. (2020) [12]	Telecom	9	This study designs a novel customer churn prediction method by using evolutionary algorithm to optimize the EMPC in the process of building a decision tree.	-	-	-	-
Maldonado et al. (2020) [27]	Telecom	9	It presents three profit-driven churn prediction methods that optimize the profit during model training.	-	-	-	Friedman test and Holm test
Jafari-Marandi et al. (2020) [30]	Telecom	1	This study proposes a profit-driven classifier that integrates customer misclassification costs into model training.	-	-	-	-
Maldonado et al. (2021) [2]	Mutual fund	2	This study develops a multi-threshold, multi-segment churn prediction framework to maximize the profit of retention campaigns.	-	-	-	-
Lessmann et al. (2021) [17]	Multiple fields	25	To maximize the profit, this study uses directed hill-climbing to select the ensemble members and averages the prediction results of the ensemble members by directed hill-climbing.	Yes	Directed hill-climbing	-	Friedman test and Rom's procedure
Janssens et al. (2022) [31]	B2B	1	It considers the difference of customer values and incorporates them into the model construction by changing the objective function of extreme gradient boosting tree.	-	-	SHAP value	-
Liu et al. (2023) [32]	Multiple fields	8	This study utilizes Bayesian methods to optimize class weights and hyperparameters in extreme gradient boosting trees for predicting customer churn.	-	-	-	Wilcoxon signed-rank test
PWC	Multiple fields	8	It proposes a profit-driven weighted classifier by integrating weighted strategy, AHA, and multiple profit-driven ensemble members.	Yes	AHA	Weighted SHAP value	Friedman test and Holm test

Previous studies on profit-driven CCP have predominantly focused on individual classifiers, with little attention paid to the potential benefits of combining multiple classifiers using weighted strategies. In this study, we propose a profit-driven customer churn classifier that incorporates the predictions of multiple profit-driven ensemble members based on a profit-driven weighted strategy. The weighted SHAP value was calculated to improve the interpretability of the PWC. The experimental results and statistical tests confirmed the effectiveness of the PWC in achieving maximum profit from retention campaigns and providing intuitive and interpretable predictions.

2.2. Research on ensemble classifiers for customer churn prediction

Researchers have focused on ensemble classifiers to improve the accuracy of CCP. Ensemble classifiers are categorized into bagging, boosting, and stacking. Bagging classifiers combine weak classifiers to reduce generalization errors [33]. They train multiple weak classifiers based on different training sets obtained by separately bootstrapping and voting on the predictions corresponding to the test set based on the predictions of all the weak classifiers, significantly reducing the variance of the predictions. Random forest [34], a typical bagging classifier, has been effectively applied to CCP [35–37].

Boosting classifiers are a type of ensemble learning method that trains weak classifiers individually in a stepwise process and combines them to construct a stronger classifier [38]. During the training stage, the boosting classifier adjusts the sample weights based on the errors of the previous weak classifiers at each iteration, allowing the next weak classifier to better handle these errors. During the prediction stage, the boosting classifier combines the predictions of multiple weak classifiers using weighted averaging to obtain the final predictions. Boosting classifiers are effective at making predictions, and several researchers have used them to study CCP [31,39–41].

The stacking classifiers train all ensemble members based on the training set. The predictions of all ensemble members corresponding to the training or validation sets are used as features to generate a new training set, which is then used to train the meta-classifier [42]. For example, Bae and Kim (2010) constructed a stacking classifier for CCP using six different classifiers, which yielded more accurate results [43]. Wong et al. (2020) developed a stacking classifier based on multiple cost-sensitive deep neural networks which gave excellent results for CCP [44]. Karuppaiah and Palanisamy (2021) employed gradient boosting trees and naïve Bayes as ensemble members and one-class SVM as meta-classifier [45].

Both bagging and boosting classifiers use homogeneous combinations of weak classifiers, while stacking classifiers use either homogeneous or heterogeneous ensemble members. Stacking classifiers with heterogeneous ensemble members can further improve the accuracy of CCP when bagging or boosting classifiers are used as ensemble members. However, stacking classifiers have a significant disadvantage in terms of their explanatory power, which is critical for making business decisions.

The theory of combined prediction suggests that weighting the results of multiple predictors can effectively enhance prediction performance [46]. Previous studies have verified the effectiveness of weighted strategies in CCP [17,47]. The most significant advantage of the weighted strategy is that it integrates the predictions of multiple predictors to effectively identify churners and non-churners. Therefore, when the bagging and boosting classifiers are used as ensemble members of a weighted classifier, the prediction performance can be further improved based on the strengths of the bagging and boosting classifiers. However, few scholars have focused on the application of weighted classifiers in CCP. Hu et al. (2020) constructed a weighted classifier using a decision tree and neural network, and the weights of these classifiers were calculated based on the probability of each sample being predicted by the decision tree and neural network [47]. Lessmann et al. (2021) used multiple classifiers as candidates and employed directed hill climbing to select and ensemble the classifiers [17]. To the best of our knowledge, these are the only two studies that have used weighted classifiers in CCP applications.

However, applying weighted classifiers to CCP poses a challenge in determining the weights in a reasonable manner. Hu et al. (2020) subjectively calculated the weights by considering only statistical indicators without considering business indicators in the CCP domain [47]. Lessmann et al. (2021) only considered whether a classifier would be profitable when added to a portfolio of classifiers [17]. If there is no gain after adding another classifier to the portfolio, the growth of the portfolio stops immediately, ignoring the possibility that more classifiers can be added in subsequent phases to bring profits. To address this issue, we used the AHA to optimize the weight coefficients with the highest profits, considering all ensemble members simultaneously when constructing the weighted classifier.

Although SHAP values can provide explanatory capabilities for most classifiers, calculating SHAP values for all ensemble classifiers requires considerable time and computational resources. Due to its properties [18], SHAP is well-suited for improving the interpretability of weighted classifiers, which weight the predictions of multiple classifiers.

Based on the features of the weighted classifier, we first calculated the SHAP values of the ensemble members, and then relied on the characteristics of the SHAP values to compute the weighted SHAP values of the PWC. Therefore, the PWC can integrate the interpretative analysis of the initial features of the predictions of each ensemble member into the interpretative analysis of the predictions of the PWC. Meanwhile, because the input features of the meta-classifier in stacking classifiers are the predictions of the ensemble members, the interpretative analysis using SHAP values can only explain the influence of the predictions of ensemble members on the predictions of the meta-classifier and not the influence of the initial features on the predictions of the meta-classifier.

3. Methodology

This section introduces the performance evaluation metrics and framework of the proposed weighted classifier.

3.1. Performance evaluation metrics

This section introduces three evaluation indicators to assess the performance of the classifiers used for CCP. These indicators evaluate the prediction accuracy and profitability of the classifiers. The CCP problem is essentially a binary classification problem, with the objective of predicting the binary label $Y \in \{0,1\}$. Y=1 indicates that the sample is classified as a churner; otherwise, it is classified as a non-churner. **Table 2** shows the confusion matrix; the prior probabilities of the non-churner and churner are set as π_0 and π_1, respectively, and their corresponding cumulative density functions are defined as F_0 and F_1, respectively. T represents the classification threshold.

Table 2
Confusion matrix

Confusion Matrix		**Predicted**	
		Churner	**Non-Churner**
Actual	**Churner**	TP $\pi_1 \times F_1(t)$	FN $\pi_1 \times (1 - F_1(t))$
	Non-Churner	FP $\pi_0 \times F_0(t)$	TN $\pi_0 \times (1 - F_0(t))$

The classifiers were evaluated using both accuracy and profit metrics, including AUC, top decile lift (TDL), and EMPC. The AUC is commonly used to evaluate the accuracy of different predictors due to its objectivity, simplicity, and intuitive interpretation. The AUC measures the ability of a predictor to distinguish between churners and non-churners based on prediction probabilities, making it well suited for dealing with unbalanced classification issues. The mathematical expression for the AUC is:

$$\boldsymbol{AUC} = \frac{1}{N_0 N_1} \sum_{i=1}^{N_1} \sum_{j=1}^{N_0} \mathbb{1}_{(p_i > p_j)} \tag{1}$$

where N_0 and N_1 represent the total number of non-churners and churners, respectively. p_i and p_j are the probability values estimated by the predictors for churners and non-churners, respectively. $\mathbb{1}$ denotes an indicator function with a value of 1 when $p_i > p_j$; otherwise, the value of $\mathbb{1}$ is 0. $AUC \in [0.5, 1]$The larger the AUC, the better the prediction accuracy.

TDL is another metric to compare the predictive performance of the predictors. Lift pays attention to those customers who are most likely to churn, and TDL (10%) calculates the ratio of the proportion ($\pi_{10\%}$) of churners in the top decile of ordered posterior churn probabilities to the proportion (π) of churners in all samples [48]: $TDL(10\%) = \pi_{10\%} / \pi$. The TDL indicator provides a valuable reference for customer retention activities, as companies can focus on minority customer groups and spend more incentives on customers who are most at risk of churn.

In addition to AUC and TDL, we used a profit-oriented performance indicator, EMPC, to evaluate the profitability of the predictive classifiers. While accuracy metrics consider the cost of misclassification as well as the benefit of correct classification, this is unrealistic in practical applications of CCP, which is an imbalanced classification problem where the minority class (i.e., churners) is the primary focus of companies. To address this issue, Verbraken et al. (2013) proposed a cost-benefit scheme that takes into account the costs associated with retention campaigns and the benefits of retaining clients [22]. They define the average profit of a retention campaign as follows:

$$\boldsymbol{P}_C\left(t, \gamma, \overline{\boldsymbol{CLV}}, \delta, \varphi\right) = \overline{\boldsymbol{CLV}} \times \left[\gamma \times (1-\delta) - \varphi\right] \times \pi_1 \times F_1 - \overline{\boldsymbol{CLV}} \times (\delta + \varphi) \times \pi_0 \times F_0 \tag{2}$$

where t is the classification threshold, γ denotes the probability that an individual is correctly identified as a churner, and will accept the incentive and abandon switching to competitors. $\overline{\boldsymbol{CLV}}$ is the average lifetime value of all clients. $\delta = d / \overline{\boldsymbol{CLV}}$ and $\varphi = f / \overline{\boldsymbol{CLV}}$, where d and f are the costs of offering incentives and contacting clients, respectively. The $\overline{\boldsymbol{CLV}}$ value is typically larger than the parameter values of d and f.

Verbraken et al. (2013) introduce a deterministic profit-based indicator, the maximum profit criterion (MPC), which aims to maximize profit by optimizing the classification threshold

t [22]. The MPC is defined as follows:

$$\boldsymbol{MPC} = \max_{\forall t}\left\{\boldsymbol{P}_C\left(t,\gamma,\overline{\boldsymbol{CLV}},\delta,\varphi\right)\right\} \tag{3}$$

The MPC is deterministic if the given parameters in the above equation are known, including γ, $\overline{\boldsymbol{CLV}}$, δ, and φ. However, the probability that a potential churner will accept incentive γ cannot be accurately estimated in a retention campaign. To this end, a probability density function for γ, that is, $h(\gamma)$, is introduced to construct a probabilistic metric, EMPC:

$$\boldsymbol{EMPC} = \int_{\gamma}\boldsymbol{P}_C\left(t^*\left(\gamma\right);\gamma,\overline{\boldsymbol{CLV}},\delta,\varphi\right)\cdot h\left(\gamma\right)d\gamma \tag{4}$$

where $t^*\left(\gamma\right)$ represents the optimal classification threshold that can maximize profit for a given γ, that is, $t^*\left(\gamma\right)=\arg\max_{\forall t}\left\{\boldsymbol{P}_C\left(t,\gamma,\overline{\boldsymbol{CLV}},\delta,\varphi\right)\right\}$.

3.2. Proposed weighted classifier based on AHA

This section introduces the modeling process of the PWC, which comprises three phases: profit-driven prediction based on ensemble members, profit-driven intelligent calculation of weight coefficients, and calculation of weighted SHAP values. **Figure 1** illustrates the framework of the PWC. We began by preprocessing the dataset and partitioning it into training, validation, and testing sets using a cross-validation mechanism. The features of the samples in the training, validation, and testing sets were normalized based on the features of the samples in the training set. We use a grid search to select the hyperparameters of the ensemble members that yield the highest profitability and use them to predict the samples in the validation and test sets. Next, we used the AHA to optimize the weight coefficients, resulting in the highest profitability based on the predictions of the weighted classifier. The weight coefficients are calculated based on the predictions of the ensemble members in the validation set. We then used the weight coefficients obtained from the AHA to calculate the PWC predictions for the test set. The predictive performance of PWC was evaluated using AUC, EMPC, and TDL (10%). Additionally, we calculated the SHAP values of the ensemble member set χ on the test set and used them to perform an explanatory analysis of the PWC based on the optimal weight coefficients.

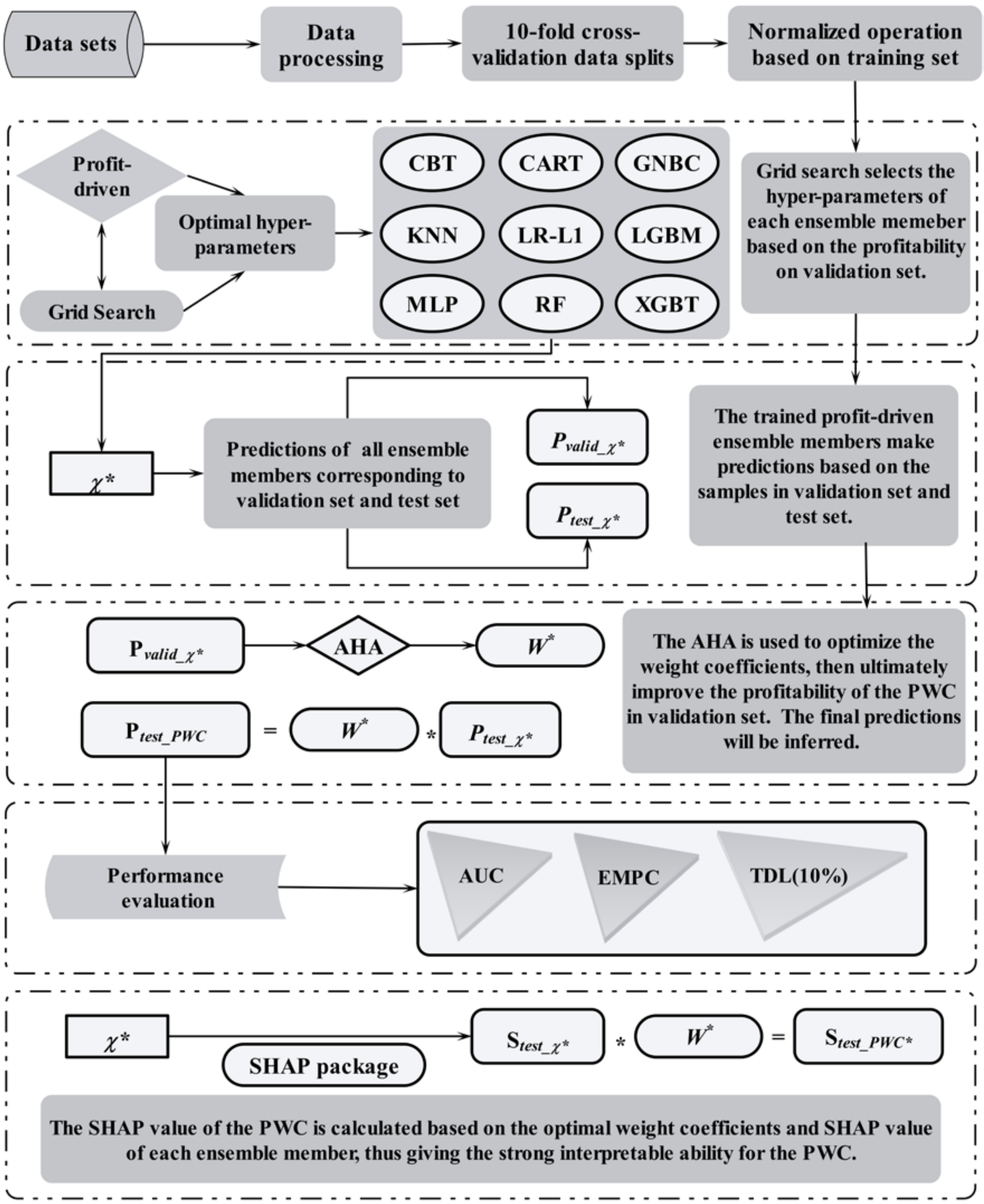

Figure 1. The framework of the PWC

3.2.1 Profit-driven prediction based on ensemble members

The CCP relies on the utilization of the classifier to determine the relationship between sample $\boldsymbol{X}$ and the binary variables $\boldsymbol{Y}$={*no churn*, *churn*}. To ensure the diversity of learning mechanisms for ensemble members, we employ multiple ensemble members, including categorical boosting (CBT) [49], classification and regression tree (CART) [50], gaussian Bayes classifier (GNBC) [51], K-nearest neighbor (KNN) [52], logistic model with L1 penalty

(LR-L1), light gradient boosting machine (LGBM) [53], multi-layer perceptron (MLP) [54], random forest (RF) [34], and extreme gradient boosting (XGBT) [55]. The set of ensemble members is denoted as

$$\chi = \{CBT, CART, GNBC, KNN, LR\text{-}L1, LGBM, MLP, RF, XGBT\} \tag{5}$$

These classifiers are denoted as $F_{\boldsymbol{Hypers}}^{\boldsymbol{classifier}}: \boldsymbol{X} \to P$, where $\boldsymbol{Hypers}$ is the hyperparameter of these classifiers. To satisfy the demands of the CCP, we adopted a grid search to select the hyperparameters of each ensemble member and determine the optimal parameters with the highest EMPC value to build profit-driven ensemble members. The objective function of the grid search for a certain ensemble member is

$$\begin{cases} \{\boldsymbol{Hypers}\} = arg \underset{\{\boldsymbol{Hypers}\}}{max} \int_{\gamma} \boldsymbol{P}_C \left(t^*(\gamma);\, \gamma, \overline{\boldsymbol{CLV}}, \delta, \varphi\right) \cdot h(\gamma)\, dc \\ s.t.\text{ The } \boldsymbol{Hypers} \text{ of each classifier have independent candidate list}, \\ \overline{\boldsymbol{CLV}} = € \ 200, d = € \ 10, f = € \ 1, \\ \delta = d \big/ \overline{\boldsymbol{CLV}}, \varphi = f \big/ \overline{\boldsymbol{CLV}}, \alpha = 6, \beta, = 14, \\ h(\gamma) = h(\gamma; \alpha, \beta) = \dfrac{\gamma^{\alpha-1}(1-\gamma)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\, du} \end{cases} \tag{6}$$

In the calculation of the EMPC, $\overline{\boldsymbol{CLV}}$ was set to € 200, d was set to € 10, f was set to € 1, $\delta = d \big/ \overline{\boldsymbol{CLV}}$, and $\varphi = f \big/ \overline{\boldsymbol{CLV}}$. In addition, the γ in EMPC is assumed to follow the beta distribution with α=6 and β=14.

Both the samples ($\boldsymbol{X}_{train}$) and the labels ($\boldsymbol{Y}_{train}$) of the training set are employed to train the classifier $F_{\boldsymbol{Hypers}}^{\boldsymbol{classifier}}$. Based on cross validation, we obtain the optimal classifier $F_{\boldsymbol{Hypers*}}^{\boldsymbol{classifier*}}$ with the optimal hyperparameters $\boldsymbol{Hypers}*$. Through grid search, we build profit-driven ensemble members set ($\chi*$) as follows:

$$\chi* = \left\{F_{\boldsymbol{Hypers*}}^{\boldsymbol{CBT*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{CART*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{GNBC*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{KNN*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{LR\text{-}L1*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{LGBM*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{MLP*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{RF*}}, F_{\boldsymbol{Hypers*}}^{\boldsymbol{XGBT*}}\right\} \tag{7}$$

The predictions of the validation set $P_{valid}^{\chi*}$ from the ensemble members were then calculated based on the samples ($\boldsymbol{X}_{valid}$) in the validation set.

$$P_{valid}^{\chi*} = \left\{P_{valid}^{\boldsymbol{CBT*}}, P_{valid}^{\boldsymbol{CART*}}, P_{valid}^{\boldsymbol{GNBC*}}, P_{valid}^{\boldsymbol{KNN*}}, P_{valid}^{\boldsymbol{LR\text{-}L1*}}, P_{valid}^{\boldsymbol{LGBM*}}, P_{valid}^{\boldsymbol{MLP*}}, P_{valid}^{\boldsymbol{RF*}}, P_{valid}^{\boldsymbol{XGBT*}}\right\} \tag{8}$$

In the process of prediction, the predictions of the test set $P_{test}^{\chi*}$ from ensemble members are also calculated based on the samples in the test set.

3.2.2 Profit-driven intelligent calculation of weighting coefficients

Based on the predictions $P_{valid}^{\chi^*}$ obtained by multiple ensemble members in the validation set, AHA was employed to optimize the weight coefficients of the ensemble members[1]. The weight coefficients are expressed as

$$\boldsymbol{W} = \left[W_{CBT}, W_{CART}, W_{GNBC}, W_{KNN}, W_{LR\text{-}L1}, W_{LGBM}, W_{MLP}, W_{RF}, W_{XGBT}\right] \tag{9}$$

The predictions of the PWC in the validation set are as follows:

$$P_{valid}^{\boldsymbol{PWC}} = \boldsymbol{W} \cdot P_{valid}^{\chi^*} \tag{10}$$

The weights were determined using the AHA, with the loss function defined as the sum of the EMPC on the validation set and an L2 penalty term calculated based on the current weights. The loss function is expressed as follows:

$$\begin{cases} \{\boldsymbol{W}\} = arg \max\limits_{\{W\}} \int_{\gamma} \boldsymbol{P}_C\left(t^*(\gamma);\, \gamma, \overline{\boldsymbol{CLV}}, \delta, \varphi\right) \cdot h(\gamma)\, dc - \theta * \|\boldsymbol{W}\|_2^2 \\ s.t.\ \boldsymbol{W}(i) = \boldsymbol{X}_{(i)} \big/ sum(\boldsymbol{X}), \boldsymbol{X} \in [0,1], i = 1,2,...,9, \\ \overline{\boldsymbol{CLV}} = € \ 200, d = € \ 10, f = € \ 1, \\ \delta = d \big/ \overline{\boldsymbol{CLV}}, \varphi = f \big/ \overline{\boldsymbol{CLV}}, \alpha = 6, \beta, = 14, \\ h(\gamma) = h(\gamma; \alpha, \beta) = \dfrac{\gamma^{\alpha-1}(1-\gamma)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\, du} \end{cases} \tag{11}$$

In Eq. (11), ***X*** denotes the solution of the AHA and the search space of the AHA for the solutions is set to [0, 1]. It is worth noting that we first transform the solutions of the AHA (***X***) to weight coefficients (***W***) so that the sum of the weight coefficients is 1 to ensure that the range of the PWC's predictions is [0, 1]. and the coefficient of L2 penalty term is set as 10^{-6}. In the calculation of the EMPC, $\overline{\boldsymbol{CLV}}$ was set to € 200, d was set to € 10, f was set to € 1, $\delta = d \big/ \overline{\boldsymbol{CLV}}$, and $\varphi = f \big/ \overline{\boldsymbol{CLV}}$. In addition, the γ in EMPC is assumed to follow the beta distribution with α=6 and β=14.

After optimization using the AHA, the optimal solution, $\boldsymbol{X}^*$ are obtained based on the validation set. To maintain the prediction probability of the PWC within [0, 1] in the subsequent stages, we fine-tuned the weight coefficients of the ensemble members and obtained the final

[1] The introduction of AHA is listed in **Appendix**.

weight coefficients $\boldsymbol{W}^*$ based on Eq. (12).

$$\boldsymbol{W}_{(i)}^* = \frac{\boldsymbol{X}_{(i)}^*}{\sum_{i=1}^{9} \boldsymbol{X}_{(i)}^*} \tag{12}$$

The predictions of the PWC for test set are calculated as:

$$P_{test}^{\boldsymbol{PWC}} = \boldsymbol{W}^* \cdot P_{test}^{\chi^*} \tag{13}$$

Finally, we obtain the final predictions of the PWC ($P_{test}^{\boldsymbol{PWC}}$). The pseudocode for using the AHA to optimize the weight coefficients is listed in **Table 3**.

Table 3

The pseudocode of using AHA to optimize weight coefficients

***Fitness functions*:** $max\left\{EMPC - \theta * \|\boldsymbol{W}\|_2^2, EMPC = \int_\gamma \boldsymbol{P}_C\left(t^*(\gamma); \gamma, \overline{\boldsymbol{CLV}}, \delta, \varphi\right) \cdot h(\gamma)\, dc\right.$

Parameters:

IterMax—max iteration number
t—current iteration number
$P_{valid}^{\chi^*}$ —the predictions matrix from ensemble members for validation set
$P_{valid}^{\boldsymbol{PWC}}$ —the predictions based on evaluated solution (weights) for validation set
$\boldsymbol{Y}_{valid}$ —the labels of samples in validation set
PopSize—population size
X—the solution of the AHA
X*—the optimal solution of the AHA
W—the weights determined based on AHA
W*—the optimal weights determined based on AHA

```
/* Initialize the population of hummingbirds (solutions) by Eq. (19). */
/* Evaluate the fitness of each hummingbird in the population. */
/* Set the best hummingbird as the best solution. */
/* Set the maximum number of iterations (IterMax). */
/* Initialize the visit table for each hummingbird. */
/* Initialize the population of hummingbirds (solutions). */
WHILE (t< IterMax) DO
/* Initialize direction vector. */
FOR each hummingbird i in the population DO
/* Generate a random number r within [0, 1]. */
IF r<1/3
/* The i-th hummingbird will conduct diagonal flight by Eq.(21). */
ELSE IF r>2/3
/* The i-th hummingbird will conduct omnidirectional flight Eq. (22). */
ELSE IF 1/3 ≤ r ≤ 2/3
/* The i-th hummingbird will conduct axial flight by Eq.(20). */
/* Use directional vector to record the flight of the i-th hummingbird. */
END IF
/* Generate another random number k within [0, 1]. */
IF k<1/2
/* The i-th hummingbird conducts guided foraging, updates the location by Eq. (24). */
/* Convert new location (solution) X to weights W by:
W_(i) = X_(i) / Σ_{i=1}^{9} X_(i) . */
/* Calculate the predictions based on W for validation set by P_valid^PWC = W · P_valid^χ* . */
/* Calculate the fitness based on current solution in validation set by EMPC − θ * ||W||_2^2 . */
```

/* (EMPC are calculated based on $P_{valid}^{\boldsymbol{PWC}}$ and $\boldsymbol{Y}_{valid}$). */

/* Compare the fitness of the new position with that of the current position. If the fitness of the new position is better, update the current position and fitness, and update the visit table. Otherwise, only update the visit table and reset the visit count of the *i*-th hummingbird at the target food location to zero. */

ELSE IF $k \geq 1/2$

/* The *i*-th hummingbird conducts territorial foraging, updates the location by Eq. (25). */

/* Convert new location (solution) $\boldsymbol{X}$ to weights $\boldsymbol{W}$ by:

$\boldsymbol{W}_{(i)} = \boldsymbol{X}_{(i)} \Big/ \sum_{i=1}^{9} \boldsymbol{X}_{(i)}$. */

/* Calculate the predictions based on $\boldsymbol{W}$ for validation set by $P_{valid}^{\boldsymbol{PWC}} = \boldsymbol{W} \cdot P_{valid}^{\chi^*}$. */

/* Calculate the fitness based on current solution in validation set by $\boldsymbol{EMPC} - \theta * \|\boldsymbol{W}\|_2^2$. */

/* (EMPC are calculated based on $P_{valid}^{\boldsymbol{PWC}}$ and $\boldsymbol{Y}_{valid}$). */

/* Compare the fitness of the new position with that of the current position. If the fitness of the new position is better, update the current position and fitness, and update the visit table. Otherwise, only update the visit table. */

END IF

END FOR

IF *t* can be divided by 2**PopSize*

/* Select the hummingbird with the optimal fitness, randomly reset the position, recalculate the fitness, and update the visit table. */

END IF

/* Select the best hummingbird and fitness value from the population. */

END WHILE

/* Obtain the optimal solution ($\boldsymbol{X}^*$) of the AHA, and transform the optimal solution ($\boldsymbol{X}^*$) as the optimal weights ($\boldsymbol{W}^*$) for the PWC. */

3.2.3 Calculation for weighted SHAP values

The SHAP value, proposed by Lundberg and Lee (2017) [18], is a valuable contribution to the development of explainable machine learning methods [56,57]. The SHAP value represents the average of the marginal contributions across all the permutations. Traditional feature importance assessments, such as the Pearson correlation, focus on measuring the relationship of the overall population while ignoring individual cases. The SHAP value successfully addresses the limitations of traditional feature-importance assessments by introducing local interpretability, which determines and compares the impact of features on a subset of instances. The SHAP values (Φ_i) were calculated as follows [18]:

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \left(\frac{|S|!\left(|N| - |S| - 1\right)!}{|N|!} \right) \left[V_{S \cup \{i\}}\left(S \cup \{i\}\right) - V_S\left(S\right) \right] \tag{14}$$

Here, *N* denotes the set of all features and *S* represents a subset of *N*. $V_{S \cup \{i\}}$ and V_S represent the methods trained with and without the *i*-th feature, respectively, and their outputs are compared based on the input features of the subset *S*.

The SHAP package [2] trained using the training set was used to construct an explainer for each ensemble member. Then, we calculated the weighted SHAP values of the PWC based on the optimal weight coefficients obtained in Section 3.1.2[3]. The SHAP values for each sample in the test set corresponding to all ensemble members were as follows:

$$\boldsymbol{S}_{test}^{\chi*} = \left\{ \boldsymbol{S}_{test}^{\boldsymbol{CBT*}}, \boldsymbol{S}_{test}^{\boldsymbol{CART*}}, \boldsymbol{S}_{test}^{\boldsymbol{GNBC*}}, \boldsymbol{S}_{test}^{\boldsymbol{KNN*}}, \boldsymbol{S}_{test}^{\boldsymbol{LR\text{-}L1*}}, \boldsymbol{S}_{test}^{\boldsymbol{LGBM*}}, \boldsymbol{S}_{test}^{\boldsymbol{MLP*}}, \boldsymbol{S}_{test}^{\boldsymbol{RF*}}, \boldsymbol{S}_{test}^{\boldsymbol{XGBT*}} \right\} \quad (15)$$

$\boldsymbol{S}_{test}^{\chi*}$ is calculated using profit-driven ensemble members ($\chi*$). Then, according to the characteristics of the SHAP value, the SHAP value of each ensemble member was adopted to calculate the weighted SHAP values of the PWC based on the optimal weight coefficients ($\boldsymbol{W}^*$) obtained by the AHA. Finally, we obtained the weighted SHAP values for each sample in the test set.

$$\boldsymbol{S}_{test}^{\boldsymbol{PWC*}} = \boldsymbol{W}^* * \boldsymbol{S}_{test}^{\chi*} \quad (16)$$

PWC possesses good interpretability owing to the weighted SHAP values, which can help explain the contribution of each feature to the prediction process.

4. Experimental setup

This section introduces the data source and experimental setup used to evaluate the predictive performance of PWC. We used eight datasets as samples in this study, and the detailed characteristics of the studied data are listed in **Table 4**. To conduct ten-fold cross-validation, we randomly removed a few samples such that the number of samples in each dataset could be divided by 10[4]. We removed ID features and features with a single value from the initial features, and conducted one-hot encoding for all category features. After treatment of all features, the final number of features ranged from 12 to 89 across datasets 1 to 8. The final churn rates ranged from 11.70% to 53.54%, indicating that the studied datasets were imbalanced, which adversely affected the predictive results. Once the training, validation, and test sets were determined by cross-validation, the values of each feature in the training set were normalized. The normalization rules of the training set were then applied to normalize the feature values of

[2] https://github.com/slundberg/shap

[3] In the **appendix**, we discuss the characteristics of weighted SHAP values to demonstrate their ability to explain the predictions of weighted classifiers. We prove that weighted SHAP values have the same characteristics as SHAP values.

[4] To perform ten-fold cross-validation, we divided the dataset into 10 equal parts. We then sequentially selected one part as the test set, randomly selected one part as the validation set from the remaining 9 parts, and used the samples other than the test and validation sets as the training set.

the samples in the validation and test sets.

Table 4
Data characteristic of the eight involved datasets

ID	Domain	Sample Number	Feature Number	Numerical Feature Number	Churn Rate (%)	Source
#1	E-Commerce	5630	19	10	16.84	Kaggle[a]
#2	Telecom	7043	28	3	26.54	Kaggle[a]
#3	Bank	10000	13	5	20.37	Kaggle[a]
#4	Newspaper	15855	18	2	19.15	Kaggle[a]
#5	Insurance	33908	15	6	11.70	Kaggle[a]
#6	Uber	50000	12	7	53.54	Kaggle[a]
#7	Telecom	51047	57	35	28.82	Kaggle[b]
#8	Telecom	71047	70	29	29.01	Kaggle[b]

a. www.kaggle.com
b. The dataset is sourced from Duke University but is no longer available on the original source page. This dataset was obtained from Kaggle (www.kaggle.com).

Table 5 presents the detailed parameter values of the ensemble members. GNBC has no preset hyperparameters, whereas the hyperparameters of the other ensemble members are determined by a grid search with the objective function of maximizing the EMPC metric. In addition, the two significant hyperparameters of AHA (population size and maximum iterations) were each set to 100.

As an existing profit-driven individual classifier, ProfLogit [26] is selected as a benchmark classifier. To highlight the rationality of the PWC ensemble strategy, we construct ensemble classifiers based on different strategies as benchmark classifiers for comparison. First, we constructed several stacking classifiers by training the ensemble members with the training set and using the predictions of the ensemble members corresponding to the validation set to train the meta-classifier. In this study, the ensemble members in the PWC were used as the ensemble members in the stacking classifiers, and all ensemble members were used as meta-classifiers to build the stacking classifiers. Second, we built a benchmark classifier based on a profit-driven selection strategy that selects the optimal ensemble member with the highest EMPC value on the validation set to predict the samples in the testing set. Third, we constructed several linear weighted classifiers based on different strategies, including an ensemble classifier based on the average weighted strategy, profit-driven rank-weighted strategy, and profit-driven probability-weighted strategy. For the ensemble classifier based on the average-weighting strategy, the weights were set to be identical. For the ensemble classifier based on the profit-driven rank-weighted strategy, we calculated the ranks of the ensemble members based on the EMPC values of the validation set and determined the weights of the ensemble members using the following formula:

$$W_i = rank_i \Big/ \sum_{k=1}^{n} rank_k \tag{17}$$

where $\boldsymbol{W}_i$ denotes the weight of the *i-th* ensemble member, and $rank_i$ denotes the rank of the *i-th* ensemble member based on the EMPC values in the validation set. For the ensemble classifier based on the profit-driven probability-weighted strategy, we determine the weights of the ensemble members based on the softmax function:

$$W_i = exp\left(EMPC_i\right) \Big/ \sum_{k=1}^{n} exp\left(EMPC_k\right) \tag{18}$$

where $EMPC_i$ denotes the EMPC value of the *i-th* ensemble member in the validation set.

In addition, we adopted an existing selective weighted classifier, namely the profit-conscious ensemble selection (PCES) [17]. To ensure a fair comparison with PWC, we replaced the ensemble members in PCES with the ensemble members of PWC and set the goal of PCES as EMPC maximization. We constructed a benchmark classifier for maximizing the statistical indicator (AUC) to verify the necessity of constructing a PWC to achieve profit. Finally, to reflect the advantages of AHA in optimizing weight coefficients, we constructed a benchmark classifier by replacing AHA with a genetic algorithm (GA). Overall, there are 26 types of benchmark classifiers. The hyperparameters in these comparative classifiers are consistent with those in the classifiers listed in **Table 5**.

Table 5
Parameter setting of AHA, GA, grid search, and classifiers

Classifier	Parameters	Candidate Values
AHA	Population size	100
	Maximum iteration	100
GA	Population size	100
	Maximum iteration	100
Grid Search	Cross validation fold	5
ProfLogit	Lambda	[0.01,0.05,0.1,0.5,0.99]
	Alpha	[0.01,0.05,0.1,0.5,0.99]
CBT	Learning rate	[0.01,0.05,0.1,0.5,1]
	Number of decision tree	50
	Depth of decision tree	[3,5,7,9]
	Minimum sample number of nodes in decision tree	[1,5,10,50,100]
CART	Max depth of decision tree	[3,5,7,9]
	Maximum proportion of randomly selected input features	[0.01,0.05,0.1,0.5,1]
	Minimum number of samples for node splitting in decision tree	[2,5,10,50,100]
	Minimum sample number of nodes in decision tree	[1,5,10,50,100]
GNBC	-	-
KNN	Number of neighbors	[1,5,10,50,100]
	Leaf size	[1,5,10,50,100]
LR-L1	Inverse of regularization strength	[0.01,0.05,0.1,0.5,1]
LGBM	Learning rate	[0.01,0.05,0.1,0.5,1]
	Number of decision tree	50
	Max depth of decision tree	[3,5,7,9]
	Number of leaves in decision tree	[2,5,10,50,100]
	Minimum sum of sample weights of nodes in decision tree	[0.01,0.05,0.1,0.5,1]
MLP	Maximum iteration	50
	Batch size	1024

	Initial learning rate	[0.01,0.05,0.1,0.5,1]
	Size of hidden layer	[50,100,150,200,250]
RF	Number of decision tree	50
	Minimum number of samples for node splitting in decision tree	[2,5,10,50,100]
	Minimum sample number of nodes in decision tree	[1,5,10,50,100]
	Max depth of decision tree	[3,5,7,9]
XGBT	Number of decision tree	50
	Learning rate	[0.01,0.05,0.1,0.5,1]
	Max depth of decision tree	[3,5,7,9]

5. Results and analyses

In this section, we investigate the following four questions: (1) Do PWC achieve better profitability than individual classifiers? (2) Are PWC superior to ensemble classifiers based on other ensemble strategies? and (3) Do PWC have interpretability? (4) Can the PWC adaptively adjust their weighting coefficients?

5.1. Do PWC achieve better profitability than individual classifiers?

The combined prediction principle provides theoretical evidence that the weighted results of different predictors outperform those of individual predictors. To determine whether the PWC outperformed the comparative classifiers, we performed 10-fold cross-validation and calculated the AUC, EMPC, and TDL (10%) values across the eight datasets. Larger metric values indicate a better predictive performance. We have marked the optimal evaluation criteria values in each dataset in bold and have provided the standard deviations in parentheses. The results in **Table 6** show that the PWC outperforms the individual classifiers in terms of prediction accuracy and profit in almost all scenarios, as demonstrated by the comparison of the AUC, EMPC, and TDL (10%) values.

To further verify the predictive effectiveness of the PWC relative to the benchmark classifiers, we followed the of statistical testing process outlined in [14] and conducted a Friedman test based on the average ranks[5] of the classifiers in terms of EMPC. The null hypothesis (H0) states that there is no significant difference in the profitability of all classifiers. Finally, we performed a Holm post hoc test to calculate the adjusted *p*-values and to avoid family-wise errors in multiple dataset comparisons. The statistical tests reported in **Table 7** confirm that the PWC generates more profit for the retention campaign than the individual classifiers. Specifically, H0 is rejected at the 0.05 significance level, indicating that the profitability of the PWC is significantly different from that of all individual classifiers. Based on the evaluation values and statistical test results, we conclude that the profitability of PWC is

[5] The average rank in our study is calculated based on 80 EMPC values obtained from ten-fold cross validation. Higher average rank denotes higher profitability.

superior to that of each individual classifier.

Table 6

AUC, EMPC, and TDL (10%) values of individual classifiers and the PWC

Indicators	AUC	EMPC	TDL (10%)	AUC	EMPC	TDL (10%)
Classifiers		**Dataset 1**			**Dataset 2**	
CBT	0.9928 (0.0057)	8.9838 (0.7540)	5.8902 (0.4015)	0.8604 (0.0135)	10.7650 (0.6773)	2.8940 (0.2605)
CART	0.9119 (0.0166)	6.9254 (0.5944)	5.0049 (0.3237)	0.8390 (0.0161)	10.3931 (0.6985)	2.7045 (0.2382)
GNBC	0.7805 (0.0284)	4.6293 (0.6079)	3.0368 (0.3189)	0.8424 (0.0116)	10.4348 (0.7567)	2.7475 (0.2181)
KNN	0.8758 (0.0255)	7.0902 (0.6504)	4.8318 (0.2156)	0.8410 (0.0117)	10.3512 (0.7660)	2.7454 (0.1769)
LR-L1	0.8185 (0.0236)	5.0191 (0.6402)	3.7255 (0.3604)	0.8451 (0.0129)	10.4163 (0.6969)	2.7370 (0.2430)
LGBM	**0.9955** (0.0031)	9.0188 (0.6778)	**5.9194** (0.3560)	0.8599 (0.0123)	10.7499 (0.6524)	2.8862 (0.2084)
MLP	0.9411 (0.0147)	7.1505 (0.7465)	5.2695 (0.2949)	0.8520 (0.0146)	10.6436 (0.7188)	2.8621 (0.2469)
RF	0.9657 (0.0092)	7.7528 (0.7253)	5.6127 (0.2744)	0.8571 (0.0126)	10.6554 (0.6761)	2.9154 (0.2715)
XGBT	0.9925 (0.0046)	8.9987 (0.6990)	5.8466 (0.3947)	0.8597 (0.0117)	10.7261 (0.6887)	2.8607 (0.2125)
ProfLogit	0.8511 (0.0242)	5.7273 (0.8137)	4.1682 (0.3087)	0.8482 (0.0133)	10.5849 (0.7288)	2.7526 (0.1966)
PWC	0.9941 (0.0053)	**9.1234** (0.6566)	5.8782 (0.3971)	**0.8605** (0.0138)	**10.8215** (0.6305)	**2.9358** (0.2532)
Classifiers		**Dataset 3**			**Dataset 4**	
CBT	**0.8677** (0.0083)	7.3062 (0.4641)	4.0491 (0.1339)	0.8717 (0.0121)	6.7616 (0.5385)	3.7809 (0.1604)
CART	0.8301 (0.0133)	6.7143 (0.5195)	3.8554 (0.1455)	0.8211 (0.0151)	5.9528 (0.5884)	3.5884 (0.1387)
GNBC	0.7848 (0.0113)	6.1127 (0.5292)	3.0872 (0.2075)	0.5869 (0.0114)	3.6528 (0.5062)	1.2937 (0.2142)
KNN	0.7681 (0.0182)	5.7125 (0.5416)	2.9404 (0.2279)	0.7623 (0.0148)	5.0098 (0.5432)	2.8456 (0.1098)
LR-L1	0.7509 (0.0164)	5.5500 (0.5735)	2.6653 (0.1959)	0.6653 (0.0172)	3.8013 (0.5342)	2.4762 (0.1576)
LGBM	0.8653 (0.0076)	7.3092 (0.4582)	4.0352 (0.1565)	0.8732 (0.0086)	6.8213 (0.5167)	3.7879 (0.1693)
MLP	0.8582 (0.0075)	7.2003 (0.5017)	3.9888 (0.1237)	0.8344 (0.0139)	6.1146 (0.4716)	3.4965 (0.1773)
RF	0.8603 (0.0100)	7.1976 (0.4513)	4.0475 (0.0859)	0.8446 (0.0124)	6.2621 (0.5876)	3.6205 (0.2048)
XGBT	0.8641 (0.0079)	7.2288 (0.4933)	**4.0960** (0.0996)	0.8723 (0.0103)	6.8039 (0.5488)	3.8217 (0.1503)
ProfLogit	0.7648 (0.0180)	6.0093 (0.5133)	2.3917 (0.1878)	0.7658 (0.0139)	5.3085 (0.5366)	2.7287 (0.1647)
PWC	0.8667 (0.0093)	**7.3470** (0.4718)	4.0492 (0.1450)	**0.8754** (0.0115)	**6.8802** (0.5780)	**3.8475** (0.2055)
Classifiers		**Dataset 5**			**Dataset 6**	
CBT	0.9315 (0.0063)	4.4485 (0.3345)	5.3947 (0.1753)	0.9421 (0.0027)	27.9855 (0.6120)	**1.8644** (0.0352)
CART	0.8836 (0.0094)	3.7079 (0.3039)	4.9533 (0.1852)	0.9058 (0.0092)	27.8595 (0.5908)	1.7923 (0.0549)
GNBC	0.8073 (0.0135)	2.6557 (0.2801)	3.9196 (0.1184)	0.7750 (0.0073)	25.2843 (0.6438)	1.4945 (0.0361)
KNN	0.8030 (0.0083)	2.5918 (0.2652)	3.9826 (0.1937)	0.8684 (0.0062)	26.8410 (0.6385)	1.7051 (0.0358)
LR-L1	0.7062 (0.0144)	1.5688 (0.1715)	3.1026 (0.1607)	0.8103 (0.0051)	26.4602 (0.6263)	1.5628 (0.0271)
LGBM	0.9314 (0.0065)	4.3862 (0.3226)	5.3384 (0.2144)	0.9439 (0.0029)	27.9800 (0.6122)	**1.8644** (0.0362)
MLP	0.9203 (0.0053)	4.1955 (0.3269)	5.1397 (0.2285)	0.8927 (0.0051)	27.8436 (0.6140)	1.7266 (0.0420)
RF	0.9205 (0.0061)	4.2129 (0.2980)	5.0661 (0.1330)	0.9115 (0.0041)	27.9032 (0.6069)	1.7874 (0.0343)
XGBT	0.9309 (0.0052)	4.4227 (0.3213)	5.3072 (0.1651)	**0.9452** (0.0017)	28.0113 (0.6168)	1.8636 (0.0355)
ProfLogit	0.8959 (0.0079)	3.7733 (0.2903)	4.8101 (0.1159)	0.8041 (0.0062)	27.8732 (0.6130)	1.4201 (0.0296)
PWC	**0.9348** (0.0058)	**4.5005** (0.3174)	**5.4155** (0.1997)	0.9440 (0.0036)	**28.0325** (0.6117)	1.8618 (0.0355)
Classifiers		**Dataset 7**			**Dataset 8**	
CBT	0.6657 (0.0068)	8.9564 (0.4023)	1.8241 (0.0704)	0.6709 (0.0031)	9.0876 (0.3752)	1.8697 (0.0532)
CART	0.6348 (0.0055)	8.7402 (0.4340)	1.5957 (0.0477)	0.6357 (0.0082)	8.8646 (0.3744)	1.6068 (0.0682)
GNBC	0.5720 (0.0090)	8.3588 (0.4483)	1.3217 (0.0735)	0.5718 (0.0082)	8.4695 (0.4331)	1.3393 (0.0676)
KNN	0.5647 (0.0076)	8.3427 (0.4378)	1.3167 (0.0822)	0.5673 (0.0049)	8.4600 (0.4251)	1.3600 (0.0472)
LR-L1	0.5000 (0.0000)	8.2710 (0.4527)	1.0255 (0.0474)	0.5000 (0.0000)	8.3942 (0.4362)	0.9830 (0.0565)
LGBM	0.6731 (0.0042)	9.0287 (0.4001)	1.8748 (0.0948)	0.6789 (0.0047)	9.1714 (0.3976)	1.9005 (0.0519)
MLP	0.6018 (0.0351)	8.5564 (0.3897)	1.4651 (0.1219)	0.6191 (0.0114)	8.7184 (0.4456)	1.5125 (0.0518)
RF	0.6582 (0.0073)	8.9414 (0.4159)	1.7676 (0.0462)	0.6582 (0.0042)	9.0488 (0.3950)	1.7502 (0.0507)
XGBT	0.6729 (0.0040)	9.0103 (0.4118)	**1.8974** (0.0583)	0.6773 (0.0039)	9.1608 (0.3846)	1.9071 (0.0627)
ProfLogit	0.5505 (0.0244)	8.5705 (0.4197)	1.1425 (0.1508)	0.5863 (0.0062)	8.6892 (0.4131)	1.3450 (0.0715)
PWC	**0.6742** (0.0040)	**9.0568** (0.4000)	1.8788 (0.0875)	**0.6791** (0.0044)	**9.1913** (0.3982)	**1.9163** (0.0613)

Table 7

Average ranks and adjusted *p*-values based on the EMPC values of the PWC and individual classifiers

Set	Classifier	Average Ranks	Adjusted *p*-value
Control	**PWC**	10.5000	-
Benchmarks	**CBT**	8.7375	0.0023*
	CART	4.9000	0.0000*
	GNBC	2.3500	0.0000*
	KNN	2.7500	0.0000*
	LR-L1	1.6313	0.0000*
	LGBM	9.1875	0.0123*

MLP	5.7438	0.0000*
RF	6.9500	0.0000*
XGBT	8.9125	0.0049*
ProfLogit	4.3375	0.0000*

a. * denotes that the null hypothesis is rejected at the significance level=0.05.

b. The Friedman statistic between the PWC and individual classifiers is 474.4949, and the *p*-value is 0.0000.

5.2. Are PWC superior to ensemble classifiers based on other ensemble strategies?

This subsection presents the construction of 16 benchmark classifiers, including classifiers based on stacking, classifier selection, average=weighted, profit-driven rank-weighted, profit-driven probability-weighted, selective ensemble, AUC-driven weighted, and profit-driven weighted strategies implemented by the GA. For the stacking classifiers, we used each ensemble member as a profit-driven meta-classifier to ensemble the predictions of all ensemble members corresponding to the validation set. The stacking classifiers included CBT-stacking, CART-stacking, GNBC-stacking, KNN-stacking, LR-L1-stacking, LGBM-stacking, MLP-stacking, RF-stacking, and XGBT stacking.

A classifier with a profit-driven selection strategy (MS-EMPC) dynamically selects the optimal classifier with the best EMPC value in the validation set. The weighted classifier with the average weighted strategy (referred to as AW-WC) calculated the final predictions based on equal weights, with all weights set to 1/9 because there were nine ensemble members. The classifier with a profit-driven rank-weighted strategy (referred to as Rank-WC) determines the weights based on the ranks of the EMPC values of the ensemble members on the validation set. The classifier with a profit-driven probability-weighted strategy (referred to as Prob-WC) transforms the weights using the softmax function based on the EMPC values of the ensemble members in the validation set.

A classifier with a profit-driven selective weighting strategy is based on an existing classifier [17] (referred to as PCES), which dynamically selects the optimal combination of ensemble members with the best EMPC value on the validation set. The weighted classifier with accuracy maximization goal (referred to as PWC-AUC) sets the goal of the grid search for ensemble members and the goal of AHA for weight optimization as AUC. The weighted classifier with optimization calculation by GA (referred to as GA-WC) sets the goal of the grid search for ensemble members and the goal of the GA for weight optimization as the EMPC.

To further verify the effectiveness of PWC, we compared it with the stacking classifiers MS-EMPC, AW-WC, Rank-WC, Prob-WC, PCES, PWC-AUC, and GA-WC. The detailed experimental results are presented in **Tables 8-9**. The PWC achieved the largest EMPC values based on six of these datasets: 10.8215 (Dataset 2), 7.3470 (Dataset 3), 6.8802 (Dataset 4), 28.0325 (Dataset 6), 9.0568 (Dataset 7), and € 9.1913 (Dataset 8). To comprehensively evaluate the application value of PWC in multiple scenarios, we performed statistical tests on the evaluation indicators across all datasets to verify whether the profitability of PWC was significantly better than that of the benchmark classifiers.

Table 10 presents the average ranks and Holm’s post hoc test results based on the EMPC of the PWC and the stacking classifiers MS-EMPC, AW-WC, Rank-WC, Prob-WC, PCES, PWC-AUC, and GA-WC. Based on the average rank and adjusted *p*-values, we observe that the PWC is significantly different from the stacking classifiers, MS-EMPC, AW-WC, Rank-WC, Prob-WC, PCES, PWC-AUC, and GA-WC, as all adjusted *p*-values are much lower than the significance level α=0.05. Although some stacking classifiers outperformed PWC in terms of profitability in certain scenarios, the profitability of PWC was significantly better in the multiple datasets application scenarios for all baseline classifiers.

Based on the experimental results, we conclude that our profit-driven weighted classifier outperforms the stacking classifiers MS-EMPC, AW-WC, Rank-WC, Prob-WC, PCES, PWC-AUC, and GA-WC, providing valuable schemes for decision-makers to maximize the profit of retention campaigns in practical applications. However, the AUC and TDL (10%) values of the PWC were not optimal in most scenarios. This is understandable and acceptable since the primary goal of a company's retention campaign is to maximize profits. In other words, achieving a satisfactory profit metric value (EMPC) is more important than achieving good AUC and TDL values (10 %).

Table 8
AUC, EMPC, and TDL (10%) values of stacking classifiers and the PWC

Indicators	**AUC**	**EMPC**	**TDL (10%)**	**AUC**	**EMPC**	**TDL (10%)**
Classifiers		**Dataset 1**			**Dataset 2**	
CBT-Stacking	0.9972 (0.0022)	9.1433 (0.6560)	5.9210 (0.3857)	0.8586 (0.0122)	10.7214 (0.6707)	2.8570 (0.2215)
CART-Stacking	0.9828 (0.0116)	8.8916 (0.7020)	5.8346 (0.3637)	0.8333 (0.0189)	10.2416 (0.8086)	2.7261 (0.3308)
GNBC-Stacking	0.9906 (0.0048)	9.0131 (0.6891)	5.4924 (0.2575)	0.8601 (0.0120)	10.7236 (0.7084)	2.8721 (0.2665)
KNN-Stacking	0.9883 (0.0097)	9.0173 (0.6502)	5.8172 (0.4745)	0.8578 (0.0107)	10.6630 (0.7287)	2.8754 (0.2166)
LR-L1-Stacking	**0.9974** (0.0020)	9.0868 (0.7135)	**5.9643** (0.3896)	**0.8610** (0.0118)	10.7463 (0.6939)	2.9307 (0.2415)
LGBM-Stacking	0.9952 (0.0065)	9.1462 (0.6730)	5.9126 (0.4234)	0.8477 (0.0119)	10.4681 (0.7442)	2.8454 (0.2024)
MLP-Stacking	0.9910 (0.0098)	9.0690 (0.7172)	5.8807 (0.4272)	0.8527 (0.0202)	10.6130 (0.8318)	2.8600 (0.2478)
RF-Stacking	0.9959 (0.0041)	9.0954 (0.6805)	5.9422 (0.3826)	0.8598 (0.0127)	10.7726 (0.6575)	2.9005 (0.3076)
XGBT-Stacking	0.9961 (0.0041)	**9.1729** (0.6823)	5.9122 (0.4131)	0.8425 (0.0154)	10.5481 (0.6977)	2.7300 (0.2069)
PWC	0.9941 (0.0053)	9.1234 (0.6566)	5.8782 (0.3971)	0.8605 (0.0138)	**10.8215** (0.6305)	**2.9358** (0.2532)
Classifiers		**Dataset 3**			**Dataset 4**	

CBT-Stacking	0.8637 (0.0104)	7.2624 (0.4827)	4.0433 (0.0890)	0.8718 (0.0109)	6.7801 (0.5430)	3.7956 (0.2245)
CART-Stacking	0.8314 (0.0171)	6.8868 (0.5258)	3.8695 (0.2165)	0.8456 (0.0134)	6.3865 (0.4489)	3.6622 (0.2605)
GNBC-Stacking	0.8600 (0.0067)	7.1882 (0.4777)	**4.1013** (0.0908)	0.8691 (0.0113)	6.7349 (0.5922)	3.8040 (0.1818)
KNN-Stacking	0.8572 (0.0092)	7.1449 (0.4725)	4.0778 (0.1183)	0.8712 (0.0113)	6.7488 (0.5505)	3.7698 (0.1782)
LR-L1-Stacking	**0.8671** (0.0082)	7.2934 (0.4940)	4.0729 (0.1119)	**0.8759** (0.0100)	6.8541 (0.5210)	3.8363 (0.1917)
LGBM-Stacking	0.8602 (0.0075)	7.1859 (0.4298)	4.0539 (0.1001)	0.8694 (0.0104)	6.7239 (0.4850)	3.7322 (0.1970)
MLP-Stacking	0.8618 (0.0104)	7.2986 (0.4982)	4.0831 (0.0919)	0.8750 (0.0103)	6.8377 (0.5485)	3.8335 (0.1786)
RF-Stacking	0.8612 (0.0107)	7.2412 (0.4740)	3.9813 (0.2049)	0.8721 (0.0113)	6.7916 (0.5567)	3.8073 (0.1586)
XGBT-Stacking	0.8561 (0.0123)	7.1721 (0.4512)	3.9596 (0.1169)	0.8685 (0.0096)	6.7267 (0.5112)	3.7394 (0.1365)
PWC	0.8667 (0.0093)	**7.3470** (0.4718)	4.0492 (0.1450)	0.8754 (0.0115)	**6.8802** (0.5780)	**3.8475** (0.2055)
Classifiers		**Dataset 5**			**Dataset 6**	
CBT-Stacking	**0.9355** (0.0054)	**4.5049** (0.3163)	5.4317 (0.1925)	0.9454 (0.0025)	27.9965 (0.6168)	1.8633 (0.0352)
CART-Stacking	0.9191 (0.0079)	4.3233 (0.3398)	5.1682 (0.2906)	0.9363 (0.0038)	27.9309 (0.5987)	1.8379 (0.0477)
GNBC-Stacking	0.9317 (0.0050)	4.4228 (0.3011)	5.2920 (0.1504)	0.9360 (0.0034)	27.9751 (0.6140)	1.8480 (0.0376)
KNN-Stacking	0.9314 (0.0070)	4.4372 (0.2947)	5.4245 (0.1738)	0.9438 (0.0028)	27.9706 (0.6207)	1.8573 (0.0346)
LR-L1-Stacking	0.9348 (0.0060)	4.4600 (0.3141)	**5.4750** (0.1795)	**0.9462** (0.0020)	28.0291 (0.6121)	**1.8659** (0.0353)
LGBM-Stacking	0.9309 (0.0110)	4.4423 (0.3203)	5.3384 (0.2523)	0.9453 (0.0023)	28.0070 (0.6105)	1.8571 (0.0287)
MLP-Stacking	0.9332 (0.0063)	4.4610 (0.3057)	5.3335 (0.2566)	0.9459 (0.0020)	28.0064 (0.6264)	1.8592 (0.0394)
RF-Stacking	0.9346 (0.0059)	4.4804 (0.3156)	5.3781 (0.2273)	0.9457 (0.0022)	28.0122 (0.6015)	1.8637 (0.0366)
XGBT-Stacking	0.9321 (0.0063)	4.4575 (0.3323)	5.3059 (0.2185)	0.9457 (0.0019)	27.9990 (0.6175)	1.8629 (0.0347)
PWC	0.9348 (0.0058)	4.5005 (0.3174)	5.4155 (0.1997)	0.9440 (0.0036)	**28.0325** (0.6117)	1.8618 (0.0355)
Classifiers		**Dataset 7**			**Dataset 8**	
CBT-Stacking	0.6718 (0.0056)	9.0161 (0.4014)	1.8815 (0.0852)	0.6795 (0.0035)	9.1653 (0.3948)	1.9089 (0.0636)
CART-Stacking	0.6492 (0.0112)	8.8254 (0.4374)	1.7324 (0.1072)	0.6628 (0.0050)	9.0333 (0.3867)	1.7534 (0.0737)
GNBC-Stacking	0.6709 (0.0048)	9.0278 (0.4066)	1.7871 (0.0801)	0.6738 (0.0036)	9.1597 (0.3823)	1.8083 (0.0476)
KNN-Stacking	0.6654 (0.0060)	8.8829 (0.4287)	1.8501 (0.0661)	0.6719 (0.0048)	9.0684 (0.3997)	1.8848 (0.0551)
LR-L1-Stacking	0.6731 (0.0042)	9.0287 (0.4001)	1.8748 (0.0948)	0.6789 (0.0048)	9.1713 (0.3975)	1.9015 (0.0515)
LGBM-Stacking	0.6724 (0.0051)	8.9952 (0.4033)	1.8783 (0.0588)	0.6784 (0.0033)	9.1505 (0.3952)	1.8984 (0.0624)
MLP-Stacking	0.6578 (0.0528)	8.9603 (0.4409)	1.8171 (0.2943)	**0.6809** (0.0035)	9.1867 (0.4008)	**1.9316** (0.0554)
RF-Stacking	0.6729 (0.0049)	9.0264 (0.4028)	**1.8838** (0.0594)	0.6794 (0.0036)	9.1662 (0.3948)	1.9101 (0.0594)
XGBT-Stacking	0.6710 (0.0040)	8.9916 (0.3919)	1.8576 (0.0604)	0.6784 (0.0032)	9.1378 (0.4153)	1.9034 (0.0574)
PWC	**0.6742** (0.0040)	**9.0568** (0.4000)	1.8788 (0.0875)	0.6791 (0.0044)	**9.1913** (0.3982)	1.9163 (0.0613)

Table 9

AUC, EMPC, and TDL (10%) values of competitive classifiers and the PWC

Indicators	**AUC**	**EMPC**	**TDL (10%)**	**AUC**	**EMPC**	**TDL (10%)**
Classifiers		**Dataset 1**			**Dataset 2**	

MS-EMPC	0.9938 (0.0056)	9.0355 (0.6824)	5.8794 (0.4027)	0.8598 (0.0132)	10.7563 (0.6278)	2.8706 (0.2111)
AW-WC	0.9920 (0.0034)	8.8753 (0.6600)	5.8346 (0.3713)	0.8598 (0.0117)	10.7097 (0.7101)	2.8666 (0.2447)
Rank-WC	0.9962 (0.0018)	9.0441 (0.6769)	5.9202 (0.3718)	0.8613 (0.0124)	10.7485 (0.6704)	2.9036 (0.2562)
Prob-WC	**0.9968** (0.0017)	9.0725 (0.6955)	5.9202 (0.3718)	0.8614 (0.0126)	10.7528 (0.6611)	2.8928 (0.2558)
PCES	0.9943 (0.0048)	9.0263 (0.6784)	5.9236 (0.4235)	0.8603 (0.0134)	10.7617 (0.6232)	2.9209 (0.2507)
PWC-AUC	0.9961 (0.0028)	9.0732 (0.7287)	**5.9323** (0.3953)	**0.8615** (0.0125)	10.7671 (0.6796)	2.9109 (0.3001)
GA-WC	0.9951 (0.0035)	9.0920 (0.6529)	5.9098 (0.3751)	0.8604 (0.0124)	10.7914 (0.6527)	2.8971 (0.2391)
PWC	0.9941 (0.0053)	**9.1234** (0.6566)	5.8782 (0.3971)	0.8605 (0.0138)	**10.8215** (0.6305)	**2.9358** (0.2532)
Classifiers		**Dataset 3**			**Dataset 4**	
MS-EMPC	0.8646 (0.0081)	7.2405 (0.4486)	4.0245 (0.1325)	0.8721 (0.0102)	6.7879 (0.5168)	3.8039 (0.1635)
AW-WC	0.8630 (0.0071)	7.2165 (0.4843)	4.0611 (0.0864)	0.8715 (0.0111)	6.7873 (0.5935)	3.8015 (0.1809)
Rank-WC	0.8673 (0.0078)	7.3036 (0.4647)	4.0532 (0.0824)	0.8750 (0.0109)	6.8283 (0.5632)	3.8292 (0.1844)
Prob-WC	**0.8677** (0.0080)	7.3143 (0.4642)	4.0430 (0.0841)	0.8755 (0.0106)	6.8373 (0.5599)	**3.8555** (0.1655)
PCES	0.8646 (0.0081)	7.2405 (0.4486)	4.0245 (0.1325)	0.8731 (0.0108)	6.7919 (0.5056)	3.8232 (0.1809)
PWC-AUC	0.8665 (0.0087)	7.2896 (0.4652)	4.0429 (0.1197)	**0.8759** (0.0107)	6.8554 (0.5353)	3.8321 (0.1805)
GA-WC	0.8665 (0.0084)	7.3320 (0.4795)	**4.0787** (0.1366)	0.8748 (0.0111)	6.8572 (0.5826)	3.8110 (0.1616)
PWC	0.8667 (0.0093)	**7.3470** (0.4718)	4.0492 (0.1450)	0.8754 (0.0115)	**6.8802** (0.5780)	3.8475 (0.2055)
Classifiers		**Dataset 5**			**Dataset 6**	
MS-EMPC	0.9314 (0.0068)	4.4214 (0.3376)	5.3719 (0.2227)	0.9441 (0.0025)	27.9945 (0.6197)	1.8640 (0.0363)
AW-WC	0.9298 (0.0055)	4.3614 (0.3055)	5.3218 (0.1477)	0.9300 (0.0039)	27.9485 (0.6165)	1.8412 (0.0363)
Rank-WC	0.9354 (0.0052)	4.4806 (0.3232)	5.4315 (0.1747)	0.9428 (0.0028)	27.9969 (0.6097)	1.8618 (0.0366)
Prob-WC	0.9358 (0.0050)	4.4865 (0.3231)	5.4319 (0.1726)	0.9412 (0.0030)	27.9852 (0.6107)	1.8603 (0.0375)
PCES	0.9327 (0.0060)	4.4585 (0.3357)	5.3733 (0.2107)	0.9452 (0.0025)	28.0106 (0.6189)	1.8648 (0.0362)
PWC-AUC	**0.9361** (0.0054)	4.4883 (0.2994)	**5.4779** (0.1841)	**0.9465** (0.0021)	28.0263 (0.6191)	**1.8655** (0.0360)
GA-WC	0.9356 (0.0050)	4.4942 (0.3085)	5.4464 (0.2081)	0.9422 (0.0048)	28.0107 (0.6102)	1.8606 (0.0343)
PWC	0.9348 (0.0058)	**4.5005** (0.3174)	5.4155 (0.1997)	0.9440 (0.0036)	**28.0325** (0.6117)	1.8618 (0.0355)
Classifiers		**Dataset 7**			**Dataset 8**	
MS-EMPC	0.6716 (0.0049)	9.0100 (0.4092)	1.8788 (0.0657)	0.6781 (0.0044)	9.1742 (0.3976)	1.9028 (0.0529)
AW-WC	0.6690 (0.0076)	9.0141 (0.4127)	1.7641 (0.0787)	0.6737 (0.0037)	9.1623 (0.3803)	1.8094 (0.0497)
Rank-WC	0.6742 (0.0054)	9.0419 (0.4148)	1.8889 (0.0855)	0.6787 (0.0036)	9.1757 (0.3811)	1.8988 (0.0504)
Prob-WC	0.6748 (0.0050)	9.0432 (0.4162)	1.9071 (0.0823)	0.6793 (0.0035)	9.1763 (0.3826)	1.9140 (0.0625)
PCES	0.6705 (0.0057)	9.0088 (0.4007)	1.8667 (0.0816)	0.6780 (0.0036)	9.1723 (0.3977)	1.8963 (0.0209)
PWC-AUC	**0.6787** (0.0043)	9.0528 (0.3935)	**1.9023** (0.0743)	**0.6813** (0.0043)	9.1888 (0.3805)	**1.9247** (0.0655)
GA-WC	0.6736 (0.0045)	9.0462 (0.4012)	1.8614 (0.0893)	0.6775 (0.0053)	9.1791 (0.3959)	1.8729 (0.0828)
PWC	0.6742 (0.0040)	**9.0568** (0.4000)	1.8788 (0.0875)	0.6791 (0.0044)	**9.1913** (0.3982)	1.9163 (0.0613)

Table 10
Average ranks and adjusted *p*-values based on EMPC values of the PWC and ensemble classifiers

Set	Classifier	Average Ranks	Adjusted p-value
Control	**PWC**	14.4938	-
Benchmarks	**CBT-Stacking**	9.7625	0.0000*
	CART-Stacking	1.9750	0.0000*
	GNBC-Stacking	6.3188	0.0000*
	KNN-Stacking	4.8875	0.0000*
	LR-L1-Stacking	10.8313	0.0000*
	LGBM-Stacking	7.6313	0.0000*
	MLP-Stacking	10.7875	0.0000*
	RF-Stacking	10.2625	0.0000*
	XGBT-Stacking	7.5063	0.0000*
	MS-EMPC	8.2188	0.0000*
	AW-WC	5.6750	0.0000*
	Rank-WC	10.4063	0.0049*
	Prob-WC	10.5438	0.0000*
	PCES	9.2313	0.0000*
	PWC-AUC	12.0750	0.0000*
	GA-WC	12.3938	0.0085*

a. * indicates that the null hypothesis is rejected at the significance level=0.05.

b. The Friedman statistic between the PWC and ensemble classifiers is 47.6561 and the p-value is 0.0000.

Table 11 presents the average running time and the average ranking based on the EMPC values, which were used to compare the trade-off between profitability and running time of the ensemble classifiers based on various strategies. For stacking classifiers that use a simple classifier as the meta-classifier (CART-Stacking, GNBC-Stacking, KNN-Stacking, LR-L1-Stacking, and MLP-Stacking), the runtime was shorter than that of PWC, but their profitability was significantly lower than that of PWC. The profitability of stacking classifiers that use an ensemble classifier as the meta-classifier (CBT-Stacking, LGBM-Stacking, RF-Stacking, and XGBT-Stacking) is generally higher than that of the aforementioned stacking classifiers; however, their runtime increases with the size of the datasets. When making predictions based on larger datasets, their operating time exceeded that of PWC, whereas their profitability did not exceed that of PWC. For ensemble classifiers based on direct selection or direct weighted strategies (MS-EMPC, AW-WC, rank-WC, Prob-WC, and PCES), the calculation time can be ignored, but their profitability is worse than that of PWC. For weighted classifiers based on intelligent optimization algorithms, PWC-AUC has a shorter runtime than PWC, and GA-WC has about the same runtime as PWC. Due to the need to calculate the driving indicator values multiple times during the process of using intelligent optimization algorithms to find the optimal weights, the running time of PWC-AUC is shorter than that of PWC with the same hyperparameters as AHA, as the running time for calculating AUC is shorter than that for calculating EMPC.

In summary, PWC achieves superior profitability in most scenarios, whereas its runtime is not significantly higher than that of the ensemble classifiers with slightly lower profitability. In some comparisons, the runtime of PWC may be several tens of seconds longer than that of other classifiers, which is insignificant in big data analysis. Moreover, as the number of predictions increases, the high profitability of PWC provides significant benefits to organizations.

Table 11
Average run time and average ranks of all ensemble strategies

Classifiers	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8
CBT-Stacking	38.20 (12.70)	41.30 (8.80)	42.50 (10.80)	42.80 (7.70)	58.50 (14.20)	81.70 (8.20)	68.30 (7.00)	57.2 (8.70)
CART-Stacking	4.00 (3.40)	4.40 (1.60)	5.00 (1.10)	5.00 (1.40)	9.00 (2.10)	10.60 (3.50)	9.90 (1.30)	11.80 (1.40)
GNBC-Stacking	0.00 (4.90)	0.00 (9.30)	0.00 (5.10)	0.00 (5.20)	0.00 (5.70)	0.00 (4.20)	0.00 (9.50)	0.00 (6.60)
KNN-Stacking	0.00 (8.10)	0.00 (5.70)	0.00 (5.00)	1.00 (6.10)	3.40 (6.50)	4.00 (4.10)	5.20 (2.00)	10.00 (1.60)
LR-L1-Stacking	0.00 (9.80)	0.00 (9.50)	0.00 (11.40)	0.00 (13.20)	0.00 (9.00)	0.00 (14.70)	0.00 (9.60)	0.00 (8.80)
LGBM-Stacking	21.00 (13.00)	24.20 (3.50)	25.70 (6.80)	28.80 (6.00)	75.30 (7.30)	161.50 (10.70)	110.40 (5.30)	116.20 (8.40)
MLP-Stacking	4.50 (8.90)	5.00 (9.40)	6.70 (11.80)	14.60 (10.60)	65.70 (10.00)	41.70 (10.80)	32.60 (12.30)	37.50 (12.50)
RF-Stacking	15.90 (11.90)	17.10 (10.90)	19.60 (8.90)	24.60 (9.70)	51.70 (10.40)	60.90 (11.00)	62.10 (9.90)	83.80 (9.40)
XGBT-	15.50 (14.40)	20.00 (4.70)	21.10 (6.40)	24.70 (4.90)	61.50 (8.20)	62.50 (9.30)	76.30 (5.30)	118.40 (6.80)

Stacking								
MS-EMPC	0.00 (7.10)	0.00 (9.80)	0.00 (7.60)	0.00 (8.60)	0.00 (5.80)	0.00 (7.70)	0.00 (8.00)	0.00 (8.60)
AW-WC	0.00 (2.40)	0.00 (7.70)	0.00 (7.20)	0.00 (8.20)	0.00 (2.20)	0.00 (2.00)	0.00 (7.90)	0.00 (7.80)
Rank-WC	0.00 (7.10)	0.00 (11.00)	0.00 (11.40)	0.00 (10.60)	0.00 (11.30)	0.00 (8.50)	0.00 (12.50)	0.50 (10.80)
Prob-WC	0.00 (9.10)	0.00 (11.00)	0.00 (12.00)	0.00 (11.00)	0.00 (11.70)	0.00 (6.00)	0.10 (12.30)	0.70 (11.20)
PCES	0.00 (7.50)	0.00 (10.30)	0.00 (7.60)	0.00 (9.50)	0.00 (8.60)	0.00 (11.00)	0.30 (8.30)	0.10 (8.90)
PWC-AUC	5.80 (8.30)	5.70 (11.00)	6.00 (11.40)	7.00 (12.80)	9.10 (11.40)	14.90 (14.10)	14.10 (13.90)	18.30 (13.70)
GA-WC	17.90 (10.70)	20.60 (13.40)	16.30 (13.20)	11.00 (12.40)	14.10 (13.50)	20.80 (11.40)	20.40 (12.60)	23.60 (11.90)
PWC	9.20 (12.70)	13.40 (14.80)	11.80 (14.30)	11.70 (14.60)	26.10 (14.80)	20.30 (15.50)	20.20 (14.70)	24.30 (14.50)

a. Outside the parentheses is the average running time of the classifiers, measured in seconds (s).

b. The values in parentheses are the average ranks of the classifiers based on the EMPC values.

5.3. Do PWC have interpretability?

In addition to the prediction performance, the interpretability of the predictive results must also be considered. The SHAP value is a useful tool for exploring the relationship between features and predictive results [58,59]. A negative SHAP value indicates that the feature has a negative influence on the predictions, while a positive SHAP value indicates that the feature has a positive influence. The SHAP value indicates the strength of the corresponding feature's influence on the predictions. Owing to space limitations, we only present the SHAP values of the PWC in Dataset 1, as shown in **Figure 2**.

Figure 2 shows the importance of all features on the predictions of the PWC, ordered from largest to smallest, with the top feature having the greatest influence on the predictions. If there are more blue dots with negative SHAP values and more red dots with positive SHAP values, the values of that feature would be positively correlated with the prediction results. In other words, customers with higher scores for this characteristic are more likely to be churners. Conversely, if there are more red dots with negative SHAP values and more blue dots with positive SHAP values, the values of this feature would be negatively correlated with the prediction results. This means that the higher the value of this feature, the lower the probability that the customer is a churner.

Figure 2 illustrates that '***Tenure***' is the most important feature that significantly impacts predictive results, with customers with shorter tenures being more likely to churn. Predictive output is also closely associated with '***Complain***', indicating that consumers are more likely to become churners if they have a complaint in the previous month. This finding highlights the importance of addressing consumer complaints in a timely manner. '***NumberOfAddress***' shows that consumers with more receiving addresses are more likely to become churners, suggesting that their consumption patterns are not stable. '***SatisfactionScore***' indicates that consumers with higher satisfaction scores are more likely to become churners[6]. This shows that enterprises should pay attention to customer satisfaction surveys; when customer satisfaction is low, they tend to churn. Characteristics that have little impact on customer churn typically affect outcomes only when their values are larger. This is a common phenomenon because many of these features are binary variables (0-1) after one-hot encoding and their combined effects with other variables can affect customer churn. For instance, if feature $\boldsymbol{X}$ is 1 and feature $\boldsymbol{Y}$ meets certain conditions, it can affect customer churn.

In summary, PWC combined with SHAP values provide powerful interpretability by providing both the strength and direction of the influence of features on predictive values.

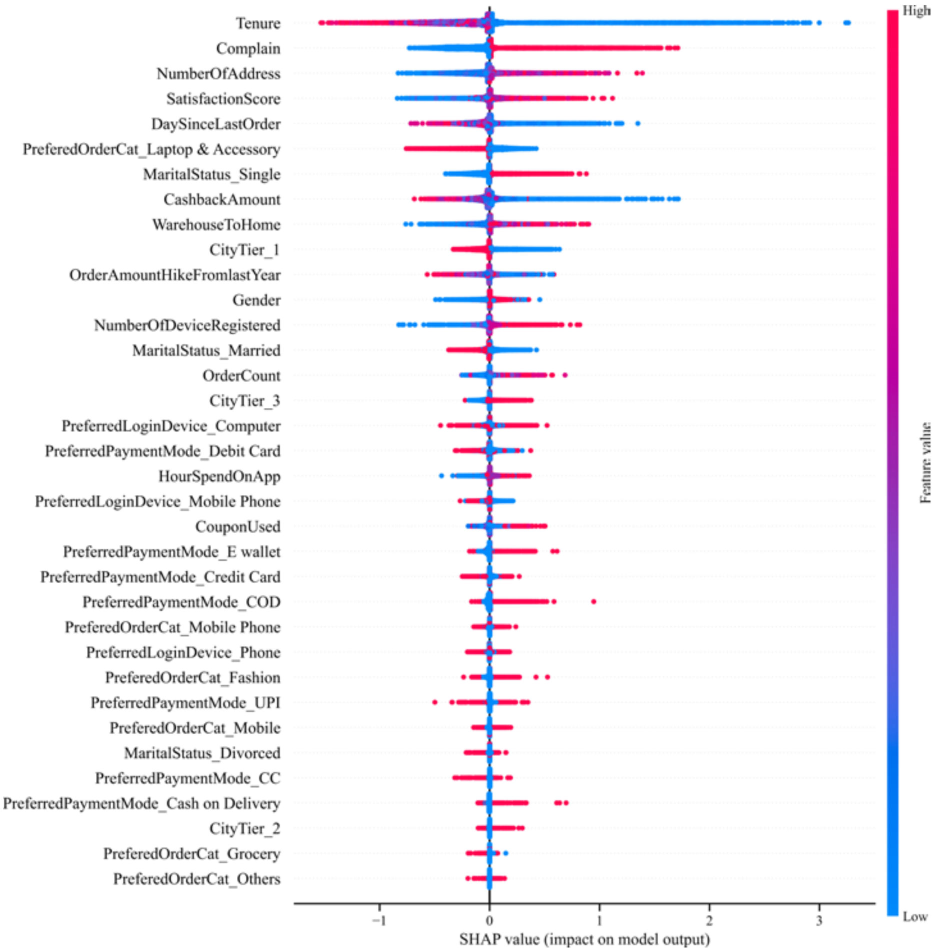

Figure 2. Visualization analysis for SHAP value in Dataset 1

[6] We have repeatedly checked the meaning of the features in Dataset 1, and the definition of '***SatisfactionScore***' is denoted as 'Satisfaction score of customers on service'. However, the uploader of the dataset did not state that the higher the satisfaction score, the more satisfied or dissatisfied with the service. Based on the understanding of common sense and the analysis results of model interpretation, we tend to prefer that the lower the satisfaction score, the higher the customer satisfaction on the service.

5.4. Can PWC adaptively adjust their weighting coefficients?

This subsection analyzes the adaptive adjustment of the weight coefficients in the PWC. **Table 12** presents the average weights of the PWC in each dataset, with the standard deviations of the weights in parentheses. The largest weight coefficient in each dataset is shown in bold, while the smallest weight coefficient is *italicized*. The table shows that the weight coefficients of the different ensemble members based on different datasets fluctuate significantly. This is because the prediction performance of different ensemble members based on different datasets is inconsistent.

Table 13 presents the EMPC values and assigned weight coefficients of the ensemble members in the PWC for Dataset 1, with the highest EMPC values and weights in bold. CBT, LGBM, and XGBT achieved higher profits and were assigned higher weights; the weights assigned to RF were the lowest compared to the other ensemble members. This may be because RF's prediction results are similar to those of the other ensemble members, which are discarded when constructing multiple regressions based on weight coefficients. The remaining ensemble members had poor profitability and were assigned lower weight coefficients. The table reveals two key findings: (1) In most scenarios, the ensemble member with the highest weight coefficient has stronger profitability. Even if an ensemble member is not the most profitable, it is still more profitable than almost all ensemble members. (2) In some scenarios, the ensemble member with the highest weight was far less profitable than the other ensemble members. However, the ensemble member with the highest profitability also has a higher weight coefficient. This indicates that the combined prediction results of these two ensemble members and others have complementary effects, leading to an improvement in PWC profitability.

Table 12 shows that the weight coefficients assigned to ensemble members by the AHA differ for different datasets. **Table 13** further shows that the PWC tends to assign higher weight coefficients to ensemble members with higher profitability or a combination of ensemble members with higher profitability. Therefore, the weight coefficients assigned to each ensemble member by the AHA are more dependent on the profitability of the ensemble members. Based on the experimental results summarized above, we confirmed that the AHA can adaptively assign optimal weight coefficients to ensemble members.

Table 12
Weight coefficients of PWC in eight datasets

Ensemble members	Dataset 1	Dataset 2	Dataset 3	Dataset 4
CBT	**0.2341** (0.1853)	0.0891 (0.0652)	**0.2569** (0.1428)	0.1999 (0.0880)
CART	0.0317 (0.0296)	0.1017 (0.0903)	0.0771 (0.0731)	0.0834 (0.0734)

GNBC	0.0618 (0.1053)	*0.0073* (0.0099)	*0.0157* (0.0243)	0.1210 (0.1013)
KNN	0.1452 (0.1306)	0.0588 (0.0800)	0.0744 (0.0790)	*0.0474* (0.0493)
LR-L1	0.0638 (0.0978)	0.0663 (0.0817)	0.0193 (0.0169)	0.0671 (0.0689)
LGBM	0.2152 (0.1766)	0.1669 (0.1639)	0.1761 (0.0977)	**0.2484** (0.1234)
MLP	0.0362 (0.0652)	**0.2111** (0.2080)	0.1554 (0.1098)	0.0804 (0.0521)
RF	*0.0089* (0.0120)	0.1053 (0.1893)	0.1018 (0.0755)	0.0609 (0.0662)
XGBT	0.2030 (0.1314)	0.1936 (0.1342)	0.1232 (0.0782)	0.0914 (0.0784)
Ensemble members	**Dataset 5**	**Dataset 6**	**Dataset 7**	**Dataset 8**
CBT	**0.2268** (0.1717)	0.2512 (0.1780)	0.0944 (0.0749)	0.1485 (0.0764)
CART	0.0267 (0.0407)	0.0200 (0.0392)	0.0870 (0.0704)	0.0326 (0.0472)
GNBC	*0.0087* (0.0104)	*0.0159* (0.0203)	*0.0240* (0.0267)	*0.0210* (0.0127)
KNN	0.1097 (0.0892)	0.0668 (0.0739)	0.0840 (0.0730)	0.0501 (0.0625)
LR-L1	0.1294 (0.1177)	0.0494 (0.0599)	0.1606 (0.0484)	0.1999 (0.0534)
LGBM	0.1334 (0.0694)	0.1856 (0.2381)	**0.2284** (0.1033)	0.1998 (0.0965)
MLP	0.1376 (0.1105)	0.0182 (0.0194)	0.1044 (0.0585)	0.0508 (0.0747)
RF	0.0545 (0.0606)	0.0238 (0.0163)	0.0686 (0.0917)	0.0604 (0.0898)
XGBT	0.1732 (0.2124)	**0.3691** (0.1457)	0.1485 (0.1067)	**0.2370** (0.1014)

Table 13
Detailed EMPC values and weight coefficients of all ensemble members in the process of ten-fold cross validation for Dataset 1

Folds	**# 1**	**# 2**	**# 3**	**# 4**	**# 5**
CBT	8.2060 (0.0061)	**9.8949** (**0.4842**)	**8.5617** (**0.3249**)	9.2081 (**0.4319**)	**10.3250** (**0.4616**)
CART	6.3592 (0.0062)	7.1931 (0.0390)	6.4777 (0.0841)	7.4506 (0.0091)	8.1051 (0.0014)
GNBC	3.5513 (0.0012)	4.7742 (0.0007)	3.9063 (0.0021)	5.2469 (0.0000)	5.2538 (0.0061)
KNN	6.9911 (0.4407)	7.5258 (0.0002)	7.2469 (0.1832)	6.7105 (0.0834)	8.0799 (0.1240)
LR-L1	4.4905 (0.0091)	5.2743 (0.0247)	4.7394 (0.0773)	5.3034 (0.0021)	6.0869 (0.1238)
LGBM	**8.4229** (**0.5301**)	9.8020 (0.0004)	8.4335 (0.0519)	**9.3152** (0.3118)	10.1155 (0.0233)
MLP	6.2936 (0.0030)	7.8150 (0.0004)	6.8358 (0.0334)	6.7674 (0.0121)	8.1510 (0.0095)
RF	7.2571 (0.0033)	8.4764 (0.0016)	7.1225 (0.0024)	8.0064 (0.0411)	9.0356 (0.0194)
XGBT	8.3375 (0.0003)	9.7564 (0.4489)	8.4243 (0.2408)	9.1744 (0.1083)	10.2665 (0.2309)
Folds	**# 6**	**# 7**	**# 8**	**# 9**	**# 10**
CBT	8.7817 (**0.3414**)	9.7962 (0.1055)	8.8227 (0.0082)	8.0604 (0.1540)	8.1815 (0.0236)
CART	6.7824 (0.0093)	7.5639 (0.0009)	6.7462 (0.0578)	6.2794 (0.0734)	6.2969 (0.0361)
GNBC	4.9227 (0.2334)	5.5416 (**0.3029**)	4.6038 (0.0018)	4.1949 (0.0477)	4.2973 (0.0220)
KNN	7.1865 (0.0437)	8.0622 (0.0002)	6.4334 (0.2822)	6.7513 (0.2037)	5.9147 (0.0905)
LR-L1	4.5677 (0.0307)	5.7638 (0.0060)	5.5394 (0.0025)	4.4642 (0.0263)	3.9616 (0.3356)
LGBM	**8.8028** (0.1428)	**9.8313** (0.2501)	9.0012 (**0.3885**)	8.0799 (0.0577)	**8.3837** (**0.3952**)
MLP	6.9131 (0.0020)	8.4975 (0.0002)	7.4476 (0.0010)	6.5348 (**0.2177**)	6.2496 (0.0830)
RF	7.6845 (0.0024)	8.5036 (0.0000)	7.8317 (0.0031)	6.8885 (0.0056)	6.7221 (0.0104)
XGBT	8.7794 (0.1942)	9.7300 (0.3342)	**9.1333** (0.2549)	**8.1386** (0.2140)	8.2468 (0.0035)

6. Conclusion and discussion

6.1. Main conclusions

This study proposes a profit-driven weighted classifier for CCP that combines profit-driven ensemble members, a profit-driven weighted strategy, and interpretable analysis. By using several widely used classifiers as ensemble members in the construction of profit-driven ensemble members the diversity of ensemble members was successfully improved. The hyperparameters of these ensemble members were determined using a grid search with the goal of profit maximization. The AHA is then innovatively employed with the objective function of maximizing the profit metric to determine the weights of the PWCs, further improving the profits of the retention campaign. Finally, the SHAP value was calculated to illustrate the relationship between the features and predictive results, providing powerful interpretability for the PWC.

Simulation experiments and statistical tests were conducted on eight datasets from multiple industries to verify the effectiveness of the PWC compared with individual classifiers and ensemble classifiers based on various ensemble strategies, including profit-driven stacking, profit-driven selection, average-weighted, profit-driven rank-weighted, profit-driven probability-weighted, profit-driven selective-weighted, AUC-driven-weighted, and profit-driven-weighted, implemented by the GA. The prediction performances of the proposed and comparative classifiers were evaluated using both accuracy metrics (AUC and TDL (10%)) and a profit metric (EMPC) in the 10-fold cross-validation process.

The experimental results demonstrate that PWC is best suited to the ultimate business objective of profit maximization. The comparative and explanatory analyses in **Sections 5.1-5.2** show that the PWC can achieve a handsome profit due to its superior weighted ensemble mechanism and excellent optimization performance of AHA. The average EMPC values of the PWC are € 9.1234, € 10.8215, € 7.3470, € 6.8802, € 4.5005, € 28.0325, € 9.0568, and € 9.1913 from datasets 1 to 8, respectively. In **Section 5.3**, the proposed classifier demonstrates strong interpretability based on the weighted SHAP value, which provides a deep exploration of the influence of the strength and direction of features on the predictive values. In addition, the weight coefficient analysis in **Section 5.4** indicates that the AHA has high adaptability in optimizing weight coefficients, suggesting that the AHA-based PWC can achieve good profitability in various scenarios. Therefore, we conclude that PWC is an effective tool that decision-makers can use in practical applications to maximize the profit of customer retention campaigns and formulate targeted policies to customer retention policies based on the contribution degree of features.

6.2. Implications

This study has two theoretical implications. The first implication pertains to the exceptional predictive performance of the proposed PWC in the pursuit of profits in the customer retention process. Future CCP studies developing heterogeneous ensemble classifiers should consider a weighted strategy based on intelligent optimization algorithms as a challenging benchmark. Second, this study provides evidence that the combination of the weighted classifier and SHAP can provide explanatory power for weighted classifiers with heterogeneous ensemble members and link it to policy making for corporate decision-makers. This case provides decision makers with the necessary information to formulate customer retention policies.

Marketing managers should direct their attention towards potential churners. Upon the indication from the proposed weighted classifier that a customer is inclined to switch to alternative products or providers, immediate action is imperative. The marketing manager should promptly engage with the customer, delivering appealing micromarketing messages, such as showcasing new products or offering more favorable prices, to proactively prevent customer churn. Thus, leveraging the proposed weighted model, marketing managers can strategically encourage the customers with a heightened likelihood of churning to remain loyal, thereby diminishing the overall churn rate. From a microcosmic perspective, the proposed weighted classifier serves as an effective tool for pinpointing potential churners and implementing preemptive measures to mitigate churn.

In the context of e-commerce, if PWC predicts that a customer is at risk of churning, businesses can implement personalized strategies to prevent customer attrition. Based on the predictions of PWC, companies can send tailored messages to potential churners, recommending products relevant to their browsing history, and offering additional discounts or free accessories as promotional incentives to enhance retention. These messages can be delivered through email, text messages, or in-app notifications to ensure timely receipt by the customer. Furthermore, if PWC identifies dissatisfaction as a potential cause for the customer churn due to specific business practices, companies can address customer concerns by adjusting these practices based on the nature of the complaints. If dissatisfaction arises from objective reasons, companies can mitigate it by modifying the discount or providing exclusive offers. This personalized interaction not only elevates customer satisfaction but also effectively mitigates potential customer churn risks.

6.3. Limitations and future research directions

Due to the limitations of the datasets, although we established the hyperparameters of the profit indicator (EMPC) based on prior research and used them to measure the profitability of classifiers, it is crucial to acknowledge that in practical scenarios, the hyperparameters of the profit indicator (EMPC) must be tailored to specific circumstances. However, the proposed PWC is driven by a profit indicator (EMPC); therefore, in practical applications, experts may be required to set the hyperparameters of the profit indicators in accordance with specific scenarios. In particular, our analysis overlooks the diversity in customer values. We explore various scenarios in which the return varies for each accepted offer, yet remains constant across all customers. Because customer spending varies in many real-world situations, it is critical to scrutinize returns that depend on individual customers. Therefore, when implementing the proposed PWC, the profit indicator should be set reasonably to make it more practical.

Going forward, three critical areas need to be explored further. First, we need to delve deeper into the reasonable determination of the hyperparameters of the EMPC to ensure the accurate computation of the profit of retention campaigns. Second, deep learning techniques should be considered to model unstructured information that can be used as an ensemble member in weighted classifiers for customer churn prediction. Finally, we should investigate more effective methods for determining the weights of weighted classifiers to enhance the predictive performance of the CCP.

Appendix

Artificial hummingbird algorithm

The AHA, a novel bio-inspired optimization algorithm proposed by Zhao, Wang, and Mirjalili (2022) [60], is mainly inspired by the flight skills, memory capacity, and foraging strategies of hummingbirds. The detailed process of the AHA is as follows:

Initialization

In the initialization process, *n* individuals are randomly assigned to *n* food sources:

$$z_i = Lb + \xi \cdot (Ub - Lb) \tag{19}$$

where z_i is the location of the *i-th* food source; Lb and Ub represent the lower and upper boundaries in the *d*-dimensional search space, respectively; and ξ denotes a random vector within [0, 1].

Flight strategies

The omnidirectional, diagonal, and axial flight strategies of individuals were employed and simulated based on a directional switch vector. The three flight strategies are then extended to a *d-D* space, where the axial flight is represented as

$$D^{(i)} = \begin{cases} 1, \ if \ \ i = randi([1,d]) \\ 0, \ else \end{cases} \quad i = 1,2,...,d \tag{20}$$

The diagonal flight is represented as:

$$D^{(i)} = \begin{cases} 1, \ if \ \ i = P(j), j \in [1,k], \\ \quad P = randperm(k), \\ \quad k \in \left[2, \left[r_1 \cdot (d-2)\right] + 1\right] \\ 0, \ else \end{cases} \quad i = 1,2,...,d \tag{21}$$

The omnidirectional flight is represented as:

$$D^{(i)} = 1, \ \ i = 1,2,...,d \tag{22}$$

where $randi([1,d])$ and $randperm(k)$ represent random integers within [1, *d*] and [1, *k*], respectively, and r_1 is a random number within [0, 1]. The diagonal flight in *d-D* space of the hummingbird is within a hyperrectangle and is bounded by any 2-to *d*-1 coordinate axes. Based on these three flight strategies, individual hummingbirds visit food sources, which generate candidate solutions.

Updated mechanism of food source location

The updated mechanism of the food source location is represented as:

$$z_i(t+1) = \begin{cases} z_i(t), & \boldsymbol{F}(z_i(t)) \le \boldsymbol{F}(v_i(t+1)) \\ v_i(t+1), & \boldsymbol{F}(z_i(t)) > \boldsymbol{F}(v_i(t+1)) \end{cases} \tag{23}$$

where $\boldsymbol{F}(\cdot)$ denotes the fitness function value.

Guided foraging

In the process of guided foraging, individuals are inclined to visit food sources that have a long unvisited time and subsequently select the one that has the highest nectar-refilling rate.

The updated individual location in guided foraging pattern is denoted as:

$$v_i(t+1) = z_{i,tar}(t) + \boldsymbol{\alpha} \cdot D \cdot \left(z_i(t) - z_{i,tar}(t)\right) \tag{24}$$

where $z_i(t)$ and $z_{i,tar}(t)$ represent the positions of the *i-th* food source and the target food source at *t*-th iteration, respectively, and $\boldsymbol{\alpha}$ denotes the guided factor that obeys the normal distribution $N(0,1)$. The updated food-source location can then be obtained using Eq. (23).

Territorial foraging

When the current food source is visited, the individual is inclined to visit a new food source in a neighboring region. The updated individual location based on the local search behavior in this pattern is denoted by

$$v_i(t+1) = z_i(t) + \boldsymbol{\beta} \cdot D \cdot z_i(t) \tag{25}$$

Here, $\boldsymbol{\beta}$ denotes the territorial factor that follows a normal distribution $N(0,1)$. The updated food-source location can then be obtained using Eq. (23).

Migration foraging

Migration foraging occurs when the individual finds that food is scarce in the area it is currently searching; in other words, when iterations exceed a preset migration factor (M=2n), the hummingbird moves from the worst food source to another random new one in the entire search space. During the migration-foraging process, the visitation table was renewed. The newly generated food source in the migration-foraging behavior is defined as

$$z_{rand}(t+1) = Lb + \boldsymbol{\xi} \cdot (Ub - Lb) \tag{26}$$

where $z_{rand}(t+1)$ represents a newly generated food source in the current iteration. The updated food-source location can then be obtained using Eq. (23).

The characteristics of weighted SHAP values

To prove that weighted SHAP values can effectively explain the predictions of weighted classifiers, we investigate whether weighted SHAP values exhibit three main characteristics: local accuracy, missingness, and consistency.

Local accuracy

Based on the characteristics of the SHAP values, we know that the SHAP values of each ensemble member satisfy the local accuracy:

$$f_j(x) = f_{base}^{(j)} + \sum_{i=1}^{n} \Phi_i^{(j)}(x) \tag{27}$$

The weighted SHAP value is calculated by weighting the SHAP values of ensemble members:

$$\Phi_i(x) = \sum_{j=1}^{m} \boldsymbol{W}_j^* \Phi_i^{(j)}(x) \tag{28}$$

To confirm that the weighted SHAP values have a local accuracy, we must prove the following equation:

$$\sum_{j=1}^{m} \boldsymbol{W}_j^* f_j(x) = f_{base} + \sum_{i=1}^{n} \Phi_i(x) \tag{29}$$

First, multiply the local accuracy equation of ensemble members by the corresponding weights:

$$\boldsymbol{W}_j^* f_j\left(x\right)=\boldsymbol{W}_j^* f_{base}^{(j)}+\sum_{i=1}^{n}\boldsymbol{W}_j^*\Phi_i^{(j)}\left(x\right) \tag{30}$$

By adding the prediction results of all ensemble members, we can obtain:

$$\sum_{j=1}^{m}\boldsymbol{W}_j^* f_j\left(x\right)=\sum_{j=1}^{m}\boldsymbol{W}_j^* f_{base}^{(j)}+\sum_{j=1}^{m}\sum_{i=1}^{n}\boldsymbol{W}_j^*\Phi_i^{(j)}\left(x\right) \tag{31}$$

For weighted classifiers, the baseline predicted value is:

$$f_{base}=\sum_{j=1}^{m}\boldsymbol{W}_j^* f_{base}^{(j)} \tag{32}$$

Therefore, the Eq. (31) can be rewritten as:

$$\sum_{j=1}^{m}\boldsymbol{W}_j^* f_j\left(x\right)= f_{base}+\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{W}_j^*\Phi_i^{(j)}\left(x\right) \tag{33}$$

Based on Eq. (28) to convert Eq. (33), Eq. (29) can be proved. This proves that the weighted SHAP values have local accuracy.

Missingness

Missingness means that if a feature value is missing, the SHAP value of the feature is zero. Since the weighted SHAP values are the weighted averages of the SHAP values of the ensemble members, if the value of the *i*-th feature is missing in each ensemble member, then the SHAP value of the *i*-th feature in each ensemble member is zero. Therefore, the weighted SHAP value of the *i*-th feature is 0. This proves that the weighted SHAP values have missingness.

Consistency

Consistency implies that, as the contribution of a feature to the predictions of the ensemble members increases, the SHAP value of that feature should also increase. Since the weighted SHAP values are the weighted averages of the SHAP values of each ensemble member, as the contribution of the *i*-th feature in each ensemble member increases, the weighted SHAP value of the *i*-th feature should also increase. This proves that the weighted SHAP values are consistent.

In summary, the weighted SHAP values exhibited local accuracy, missingness, and consistency. These properties make the weighted SHAP values a reliable method for measuring the feature importance of weighted classifiers.

Reference

[1] Oskarsdottir M, Bravo C. Multilayer network analysis for improved credit risk prediction. Omega 2021;105:102520. https://doi.org/10.1016/j.omega.2021.102520.

[2] Maldonado S, Domínguez G, Olaya D, Verbeke W. Profit-driven churn prediction for the mutual fund industry: A multisegment approach. Omega 2021;100:102380. https://doi.org/10.1016/j.omega.2020.102380.

[3] (Bill) Tseng T-L, Huang C-C. Rough set-based approach to feature selection in customer relationship management. Omega 2007;35:365–83. https://doi.org/10.1016/j.omega.2005.07.006.

[4] Gattermann-Itschert T, Thonemann UW. How training on multiple time slices improves performance in churn prediction. Eur J Oper Res 2021;295:664–74. https://doi.org/10.1016/j.ejor.2021.05.035.

[5] De Caigny A, Coussement K, De Bock KW. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. Eur J Oper Res 2018;269:760–72. https://doi.org/10.1016/j.ejor.2018.02.009.

[6] Ganesh J, Arnold MJ, Reynolds KE. Understanding the Customer Base of Service Providers: An Examination of the Differences between Switchers and Stayers. J Mark 2000;64:65–87. https://doi.org/10.1509/jmkg.64.3.65.18028.

[7] Van den Poel D, Larivière B. Customer attrition analysis for financial services using proportional hazard models. Eur J Oper Res 2004;157:196–217. https://doi.org/10.1016/S0377-2217(03)00069-9.

[8] Chen ZY, Fan ZP, Sun M. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. Eur J Oper Res 2012;223:461–72. https://doi.org/10.1016/j.ejor.2012.06.040.

[9] Lalwani P, Mishra MK, Chadha JS, Sethi P. Customer churn prediction system: a machine learning approach. Computing 2021;104:271–94. https://doi.org/10.1007/s00607-021-00908-y.

[10] De Caigny A, Coussement K, De Bock KW, Lessmann S. Incorporating textual information in customer churn prediction models based on a convolutional neural network. Int J Forecast 2020;36:1563–78. https://doi.org/10.1016/j.ijforecast.2019.03.029.

[11] Jiang P, Liu Z, Zhang L, Wang J. Hybrid model for profit-driven churn prediction based on cost minimization and return maximization. Expert Syst Appl 2023;228:120354. https://doi.org/10.1016/j.eswa.2023.120354.

[12] Höppner S, Stripling E, Baesens B, Broucke S vanden, Verdonck T. Profit driven decision trees for churn prediction. Eur J Oper Res 2020;284:920–33. https://doi.org/10.1016/j.ejor.2018.11.072.

[13] Senthan P, Rathnayaka RMKT, Kuhaneswaran B, Kumara BTGS. Development of churn prediction model using XGBoost - Telecommunication Industry in Sri Lanka. 2021 IEEE Int. IOT, Electron. Mechatronics Conf., 2021, p. 1–7. https://doi.org/10.1109/IEMTRONICS52119.2021.9422657.

[14] De Bock KW, De Caigny A. Spline-rule ensemble classifiers with structured sparsity regularization for interpretable customer churn modeling. Decis Support Syst 2021;150:113523. https://doi.org/10.1016/j.dss.2021.113523.

[15] Wezel, van M, Potharst R. Improved customer choice predictions using ensemble methods. Eur J Oper Res 2007;181:436–52. https://doi.org/10.1016/j.ejor.2006.05.029.

[16] Idris A, Khan A, Lee YS. Intelligent churn prediction in telecom: Employing mRMR feature selection and RotBoost based ensemble classification. Appl Intell 2013;39:659–72. https://doi.org/10.1007/s10489-013-0440-x.

[17] Lessmann S, Haupt J, Coussement K, De Bock KW. Targeting customers for profit: An ensemble learning framework to support marketing decision-making. Inf Sci (Ny) 2021;557:286–301. https://doi.org/10.1016/j.ins.2019.05.027.

[18] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, p. 4768–77. https://doi.org/10.5555/3295222.3295230.

[19] Glady N, Baesens B, Croux C. Modeling churn using customer lifetime value. Eur J Oper Res 2009;197:402–11. https://doi.org/10.1016/j.ejor.2008.06.027.

[20] Verbeke W, Dejaeger K, Martens D, Hur J, Baesens B. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. Eur J Oper Res 2012;218:211–29. https://doi.org/10.1016/j.ejor.2011.09.031.

[21] Elkan C. The foundations of cost-sensitive learning. Int Jt Conf Artif Intell 2001;17:973–8.

[22] Verbraken T, Verbeke W, Baesens B. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. IEEE Trans Knowl Data Eng 2013;25:961–73. https://doi.org/10.1109/TKDE.2012.50.

[23] Óskarsdóttir M, Baesens B, Vanthienen J. Profit-Based Model Selection for Customer Retention Using Individual Customer Lifetime Values. Big Data 2018;6:53–65. https://doi.org/10.1089/big.2018.0015.

[24] Verbraken T, Verbeke W, Baesens B. Profit optimizing customer churn prediction with Bayesian network classifiers. Intell Data Anal 2014;18:3–24. https://doi.org/10.3233/IDA-130625.

[25] Maldonado S, Flores Á, Verbraken T, Baesens B, Weber R. Profit-based feature selection using support vector machines - General framework and an application for customer retention. Appl Soft Comput J 2015;35:740–8. https://doi.org/10.1016/j.asoc.2015.05.058.

[26] Stripling E, vanden Broucke S, Antonio K, Baesens B, Snoeck M. Profit maximizing logistic model for customer churn prediction using genetic algorithms. Swarm Evol Comput 2018;40:116–30. https://doi.org/10.1016/j.swevo.2017.10.010.

[27] Maldonado S, López J, Vairetti C. Profit-based churn prediction based on Minimax Probability Machines. Eur J Oper Res 2020;284:273–84. https://doi.org/10.1016/j.ejor.2019.12.007.

[28] Zhang Z, Wang R, Zheng W, Lan S, Liang D, Jin H. Profit Maximization Analysis Based on Data Mining and the Exponential Retention Model Assumption with Respect to Customer Churn Problems. Proc. - 15th IEEE Int. Conf. Data Min. Work. ICDMW 2015, 2016. https://doi.org/10.1109/ICDMW.2015.84.

[29] Lee E, Kim B, Kang S, Kang B, Jang Y, Kim HK. Profit Optimizing Churn Prediction for Long-term Loyal Customer in Online games. IEEE Trans Games 2018;12:41–53. https://doi.org/10.1109/TG.2018.2871215.

[30] Jafari-Marandi R, Denton J, Idris A, Smith BK, Keramati A. Optimum profit-driven churn decision making: innovative artificial neural networks in telecom industry. Neural Comput Appl 2020;32:14929–62. https://doi.org/10.1007/s00521-020-04850-6.

[31] Janssens B, Bogaert M, Bagué A, Van den Poel D. B2Boost: instance-dependent profit-driven modelling of B2B churn. Ann Oper Res 2022. https://doi.org/10.1007/s10479-022-04631-5.

[32] Liu Z, Jiang P, De Bock KW, Wang J, Zhang L, Niu X. Extreme gradient boosting trees with efficient

Bayesian optimization for profit-driven customer churn prediction. Technol Forecast Soc Change 2024;198:122945. https://doi.org/10.1016/j.techfore.2023.122945.

[33] Breiman L. Bagging predictors. Mach Learn 1996;24:123–40. https://doi.org/10.1007/BF00058655.

[34] Breiman L. Random forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.

[35] Effendy V, Adiwijaya, Baizal ZKA. Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest. 2014 2nd Int. Conf. Inf. Commun. Technol., 2014, p. 325–30. https://doi.org/10.1109/ICoICT.2014.6914086.

[36] Idris A, Rizwan M, Khan A. Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. Comput Electr Eng 2012;38:1808–19. https://doi.org/10.1016/j.compeleceng.2012.09.001.

[37] Xie Y, Li X, Ngai EWT, Ying W. Customer churn prediction using improved balanced random forests. Expert Syst Appl 2009;36:5445–9. https://doi.org/10.1016/j.eswa.2008.06.121.

[38] Freund Y, Schapire RE. Experiments with a New Boosting Algorithm. 1996.

[39] Sharma T, Gupta P, Nigam V, Goel M. Customer Churn Prediction in Telecommunications Using Gradient Boosted Trees. Int. Conf. Innov. Comput. Commun., Singapore: Springer Singapore; 2020, p. 235–46. https://doi.org/10.1007/978-981-15-0324-5_20.

[40] Vakeel A, Vantari NR, Reddy SN, Muthyapu R, Chavan A. Machine Learning Models for Predicting and Clustering Customer Churn Based on Boosting Algorithms and Gaussian Mixture Model. 2022 Int. Conf. Adv. Technol., 2022, p. 1–5. https://doi.org/10.1109/ICONAT53423.2022.9725957.

[41] Lv P, Wu Q, Xu J, Shu Y. Stock Index Prediction Based on Time Series Decomposition and Hybrid Model. Entropy 2022;24. https://doi.org/10.3390/e24020146.

[42] Wolpert DH. Stacked generalization. Neural Networks 1992;5:241–59. https://doi.org/10.1016/S0893-6080(05)80023-1.

[43] Bae JK, Kim J. Integration of heterogeneous models to predict consumer behavior. Expert Syst Appl 2010;37:1821–6. https://doi.org/10.1016/j.eswa.2009.07.012.

[44] Wong ML, Seng K, Wong PK. Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. Expert Syst Appl 2020;141:112918. https://doi.org/10.1016/j.eswa.2019.112918.

[45] Karuppaiah KS, Palanisamy NPG. Heterogeneous ensemble stacking with minority upliftment (HESMU) for churn prediction on imbalanced telecom data. Mater Today Proc 2021. https://doi.org/10.1016/j.matpr.2020.12.893.

[46] Bates JM, Granger CWJ. The Combination of Forecasts. J Oper Res Soc 1969;20:451–68. https://doi.org/10.1057/jors.1969.103.

[47] Hu X, Yang Y, Chen L, Zhu S. Research on a Customer Churn Combination Prediction Model Based on Decision Tree and Neural Network. 2020 IEEE 5th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2020, 2020. https://doi.org/10.1109/ICCCBDA49378.2020.9095611.

[48] De Bock KW, Van den Poel D. Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. Expert Syst Appl 2012;39:6816–26. https://doi.org/10.1016/j.eswa.2012.01.014.

[49] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. Catboost: Unbiased boosting with categorical features. Adv. Neural Inf. Process. Syst., 2018, p. 6639–49. https://doi.org/10.5555/3327757.3327770.

[50] Gordon AD, Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Biometrics 1984;40:874. https://doi.org/10.2307/2530946.

[51] Domingos P, Pazzani M. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Mach Learn 1997;29:103–30. https://doi.org/10.1023/a:1007413511361.

[52] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 1992;46:175–85. https://doi.org/10.1080/00031305.1992.10475879.

[53] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst., vol. 2017- Decem, 2017, p. 3147–55.

[54] Haykin S. Neural Networks: A Comprehensive Foundation (3rd Edition). Knowl Eng Rev 1999;13:409–12. https://doi.org/10.1017/S0269888998214044.

[55] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, p. 785–94. https://doi.org/10.1145/2939672.2939785.

[56] Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat Biomed Eng 2018;2:749–60. https://doi.org/10.1038/s41551-018-0304-0.

[57] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, et al. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2020;2:56–67. https://doi.org/10.1038/s42256-019-0138-9.

[58] Zhang L, Wang J, Liu Z. What should lenders be more concerned about? Developing a profit-driven loan default prediction model. Expert Syst Appl 2023;213:118938. https://doi.org/10.1016/j.eswa.2022.118938.

[59] Liu Z, Jiang P, Wang J, Du Z, Niu X, Zhang L. Hospitality order cancellation prediction from a profit-driven perspective. Int J Contemp Hosp Manag 2023;35:2084–112. https://doi.org/10.1108/IJCHM-06-2022-0737.

[60] Zhao W, Wang L, Mirjalili S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 2022;388:114194. https://doi.org/10.1016/j.cma.2021.114194.