Science Essence Journal Vol. 41 No. 2 (2025) | 180-201

Research Article

Unveiling Customer Insights: An Interpretable Machine Learning Approach to Bank Telemarketing Data

Chin Lertvipada[1] and Sirisup Laohakiat[2*]

Received: 10 March 2025
Revised: 21 July 2025
Accepted: 29 July 2025

ABSTRACT

Financial institutions play a vital role in driving the economy. Despite the advent of digital financial systems, phone-based product offerings remain popular in the banking sector. This study focuses on building a model to predict customer applications from telemarketing campaigns. By utilizing the publicly available Bank Marketing Data Set, which exhibits significant class imbalance, we explored various combinations of effective imbalanced treatments and categorical encodings in conjunction with machine learning models to identify the most optimal combination for prediction. Additionally, interpretable machine learning techniques were employed to delve into the critical features and the underlying reasoning behind the model's predictions. The experiment revealed that the LightGBM model with Class weight and One-hot encoding yielded the best AUC score of 0.948. Using SHAP to explain the model's behavior, we found that the features related to economic factors hold greater significance compared to individual customer attributes. Furthermore, error analysis on false negative instances demonstrated that the similarity of instance characteristics of some important features could mislead the models and result in inaccurate prediction. These findings shed light on the model's decision-making process and offer insights for enhancing prediction accuracy and understanding customer behavior in financial product applications. The results offer actionable guidance for optimizing business operations by enabling more efficient lead targeting, reducing resource waste in telemarketing efforts, and supporting data-driven decision-making in customer outreach strategies.

Keywords: Machine learning, Interpretable machine learning, Bank telemarketing, SHAP

[1] Whitelabel and RTA, Agoda, Bangkok 10330, Thailand

[2] Department of Computer Science, Faculty of Science, Srinakharinwirot University, Bangkok 10110, Thailand

*Corresponding author, email: sirisup@g.swu.ac.th

Introduction

Financial institutions are pivotal to economic growth, and the ability to identify, attract, and retain depositors has long been a benchmark of operational success. Over the past decade, customer analytics has transformed this process—shifting marketing decisions from intuition-driven initiatives to data-driven, personalized outreach that spans segmentation, churn prediction, lifetime-value estimation, and campaign optimization. Within this broader landscape, predicting telemarketing responses remains an area of high practical value, because direct phone contact is still one of the most cost-effective acquisition channels for many banks.

Despite a gradual migration toward digital self-service platforms, outbound calls continue to deliver strong conversion rates—provided that calls are placed to customers who are both eligible and receptive. Unsolicited calls to low-propensity customers, on the other hand, incur monetary costs, waste staff time, and risk damaging brand reputation. Consequently, recent research has focused on machine-learning methods that can score a lead list before dialing begins, thereby aligning telemarketing with the larger movement toward precision marketing in customer analytics.

Previous telemarketing studies [1-8] have shown that tree-based ensemble models outperform traditional logistic regression when all available features are used. Yet two key gaps persist. First, many of these studies rely on information that becomes available only after the first contact—such as "call duration"—rendering their models impractical for pre-call lead screening. Second, most implementations are black boxes that offer little insight into why a particular customer is predicted to accept or decline an offer, which is increasingly important in regulated industries and customer-centric business cultures that demand transparency.

Parallel to these modeling efforts, the field of interpretable (or explainable) machine learning has matured rapidly. In account for gaining more understanding on the model, Xie et al. [9] adopt partial dependence plot (PDP) to reveal the dependency between each variable and the final decision of the customers by showing the average marginal effect of a feature across all observations. While PDPs can be useful for understanding the average effect of a feature [10], it lacks the capability to get the insight into individual level. Techniques such as SHapley Additive exPlanations (SHAP) [11] make it feasible to quantify and visualize how each feature influences a model's output—at both the global (model-level) and local (individual-customer) scale. Leveraging such methods not only satisfies regulatory and ethical requirements but also helps campaign managers refine targeting rules, craft tailored scripts, and detect bias or drift.

Numerous studies have adopted SHAP as a powerful tool to unravel the complex inner workings of machine learning models. By incorporating SHAP into their modeling pipelines, researchers have been able to delve deeper into the decision-making processes of these models, uncovering the specific factors that influence their predictions. For example, Hu et al. [12] created predictive models for the mortality of patients with acute kidney injury (AKI) who were admitted to the intensive care unit (ICU). The focus was on interpreting the models to be reliable and understandable, making the results to be

used for treatment decisions effectively. The most accurate model was the XGBoost with an accuracy of 89%. The SHAP was used to explain the model at both the model level and individual level. Similarly, Liu et al. [13] created predictive models to identify individual risk of Parkinson's disease. Feature selection methods, including F-Score, Anova-F, Mutual Information (MI), and SHAP, were utilized to create a lean model while maintaining high performance.

Building on these developments, this paper makes three main contributions to the customer-analytics and telemarketing literature:

1. **Two-stage modeling framework**: We design separate predictive models for (i) the pre-contact stage, using only information available before the first call, and (ii) the post-contact stage, where additional interaction-specific features can be exploited.
2. **Comprehensive pipeline evaluation**: We systematically compare categorical-encoding schemes (One-Hot, BaseN, CatBoost) and class-imbalance treatments (class weighting, random undersampling, SMOTE) across four widely used algorithms (Logistic Regression, Random Forest, XGBoost, and LightGBM) to identify the most robust combination.
3. **Interpretable insights with SHAP**: We pair the best models with SHAP analysis to reveal which economic, temporal, and customer-level factors drive acceptance or rejection, and we use error analysis to diagnose false-negative predictions that could impede campaign performance.

The remainder of this paper is structured as follows: Section 2 outlines the methodology, detailing the data analysis process, preprocessing steps, approaches to handling class imbalance, categorical feature encoding strategies, model development, and interpretability techniques. Section 3 presents the experimental results along with a discussion of model performance and key findings. Finally, Section 4 concludes the study by summarizing the contributions and proposing future research directions, particularly in enhancing model performance and practical applicability.

Materials and Methods

The overall approach adopted in this study is illustrated in Figure 1. The process begins with data preparation, where categorical variables are encoded into numerical formats. The dataset is then split into training and test sets for model development and evaluation. We conduct two sets of experiments: one utilizing all available features and another excluding the "duration" feature to simulate pre-contact scenarios. Finally, we apply SHAP to interpret the decision-making processes of the models and gain insights into the contribution of individual features.

Data set

This study utilizes the publicly available Bank Marketing Data Set [14], which was collected from a Portuguese bank as part of its telemarketing efforts to promote term deposit accounts. Specifically, we employed the bank-additional.zip version of the dataset, which includes a comprehensive set of

features describing individual customers and relevant socio-economic indicators. The dataset contains 41,188 instances and 20 features, grouped into three main categories: customer demographics, contact history, and macroeconomic conditions. The target variable, "y", is binary, indicating whether a client subscribed to a term deposit ("yes") or not ("no"). Notably, the dataset is highly imbalanced, with 88.7% (36,548 instances) labeled as "no" and only 11.3% (4,640 instances) labeled as "yes". A detailed description of all features is provided in Table 1.

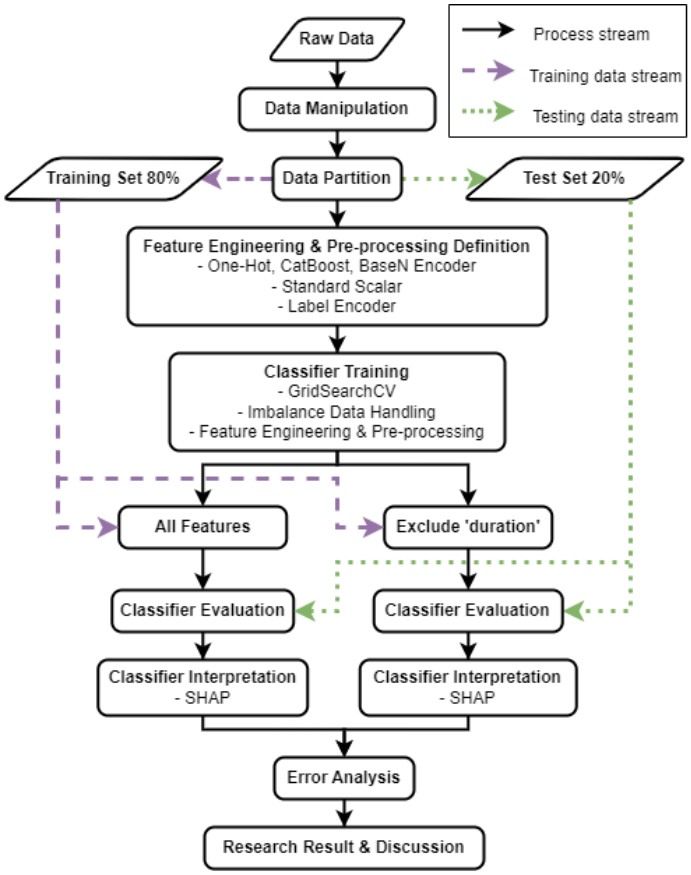

Figure 1 Experimental Procedure.

As shown in Table 1, the feature "duration" reflects the outcome of the most recent interaction regarding the customer's application status. This information is only available after initiating contact with the customer, meaning it cannot be known beforehand. Consequently, any model that relies on this feature would require randomly contacting customers first, which undermines the goal of efficient pre-screening. This limitation reduces the model's practicality in real-world telemarketing scenarios where early-stage lead selection is critical.

The original "pdays" feature, which records the number of days since the last contact with the customer, is a numerical variable. However, most entries are marked as 999, indicating no prior contact, while the remaining values (ranging up to 29) reflect recent interactions. To reduce sparsity and improve

interpretability, we transformed "pdays" into a binary feature: "Contacted Recently" (pdays < 999) and "Not Contacted Recently" (pdays = 999). This simplification helps maintain the essential information while reducing model complexity.

Table 1 Description of Bank Marketing Dataset Features.

Bank client data			**Contact information**			**Socio-economic**		
Name	**Description**	**Types**	**Name**	**Description**	**Types**	**Name**	**Description**	**Types**
age	age of customer	num	contact	contact communication type	cat	emp.var.rate	monthly average employment variation rate	num
job	type of client job	cat	month last contact	month of year	cat	cons.price.idx	monthly average consumer price index	num
marital	marital status	cat	day of week last contact	day of the week	cat	euribor3m	daily three month euribor rate	num
education		cat	duration	last contact duration (in sec.)	num	nr.employed	quarterly average of the total number of employed citizens	num.
default	has credit in default?	cat	campaign	number of contacts performed during this campaign				
housing	has housing loan?	cat	pdays	number of days since last contact (previous campaign)	num			
loan	has personal loan?	cat	previous	number of contacts in previous campaigns	num			
			poutcome	outcome of the previous campaign	cat			

Note: cat and num indicate categorical and numerical features respectively.

Feature Engineering and Data Preparation

Numeric features were processed using the Standard Scaler to achieve a normal distribution. The missing values were imputed with the mean value. For categorical features, three categorical encoding methods including, One-Hot Encoder (OH), CatBoost Encoder (CB) [15], and BaseN Encoder (BN) were employed to compare the performances for model creation.

One-hot encoder creates a binary column for each unique category in the categorical variable. If a particular instance belongs to that specific category, the corresponding binary column is set to 1, while remaining binary columns are set to 0. As a result, a variable with n-values is expanded into n variables. Note that in this method, the number of encoded features for one variable is equal to the number of values of the original feature. When employing the BaseN encoder, we utilized a different approach for encoding categorical variable values. Unlike One-hot encoding where we generate binary columns equal to the number of the values in the feature, we represent the values using a base-2 numbering system. It's worth noting that we have the flexibility to use any base number to represent the categorical feature. For instance, if we use a base-10 numbering system, this method would be equivalent to ordinal encoding.

Catboost encoding is a variant of Target encoding that differs from One-hot or BaseN encoding methods by incorporating the labels during the encoding process. To perform Catboost encoding, the instances in the dataset are first randomly shuffled. Then, the encoded value is calculated using equation (1) as follows:

$$v = \frac{(C_{in}+p)}{C_{total}} \quad (1)$$

where v indicates the encoded value, C_{in} denotes the number of times the label value was equal to 1 for objects with the current categorical feature value, C_{total} refers to the total count of objects, up to the current instance, that possess a categorical feature value matching the current one. Finally, p represents the preliminary value for the numerator determined based on the initial parameters.

The Ordinal Encoder was employed to transform ordinal values into numerical values with rank. The missing values for nominal and ordinal features were replaced with the most frequent value. The Label Encoder was employed to convert the results, "yes" and "no", to 1 and 0, respectively.

The Pipeline and Column Transformer techniques were implemented to ensure the appropriated process and prevented data leakage. These techniques facilitate the sequential execution of data preprocessing steps and ensure the appropriate handling of different types of features within the dataset.

Models Creation and Evaluation

In this study, we utilized four different models to train on our dataset: Logistic Regression, Random Forest, XGBoost [16], and LightGBM [17]. The rationale behind selecting these specific models is as follows. Firstly, we included Logistic Regression due to its simplicity and linear nature, with the expectation that it would serve as a benchmark for comparison. Additionally, the remaining three models, namely Random Forest, XGBoost, and LightGBM, were chosen because they are ensemble tree-based models known for their high performance on tabular data [18].

Logistic Regression (LR) is a widely used model due to its simplicity. Based on linear regression equation, LR performs binary classification by using logistic function to map the decision value to a probability between 0 and 1 as in equations (2) and (3).

$$d = \sum w^T x \tag{2}$$

$$\hat{y} = \frac{1}{1+e^{-d}} \tag{3}$$

where w is the coefficient of the regression model, x is the feature vector, d denotes the decision value, and $\hat{y}$ represents the predicted class probability which has value between 0 and 1. To determine the coefficient w, we minimize the regularized log-loss function as in equation (4).

$$\underset{w}{\operatorname{argmin}}\, C \sum_{i=1}^{N} \log\left(\exp\left(-y_i\left(w^T x_i\right)+1\right)\right)+\left\|w\right\|_n \tag{4}$$

where N is the number of training instances, C indicates regularization coefficient, and ‖w‖_n denotes n-norm of the regression coefficient. For LR model, we have C as the hyperparameter of the model which can be varied to find the optimal value.

Random Forest (RF) is an ensemble tree-based model which employs bagging technique to construct multiple decision trees from a single dataset by performing sampling with replacement. To create diversity among the decision trees, RF employs a random feature projection approach in which a subset of random selected features is used to train each decision tree. In this study, we use the depth of the decision tree and the number of random features as the hyperparameter for the model.

XGBoost and LightGBM are both ensemble gradient boosting tree-based models. Described as ensemble gradient boosting models, both models use a combination of decision trees, sometimes called stumps, to iteratively minimize a loss function and improve the overall performance of the model. Each decision tree is built to correct the errors of previous trees.

Although utilizing the same gradient boosting technique, both models are different in detail. XGBoost employs a level-wise tree growth strategy, where it builds trees in a depth-wise manner. On the other hand, LightGBM adopts a leaf-wise tree growth approach, where it grows tree leaf-wise, prioritizing more informative leaves. LightGBM also adopts a histogram-based approach to binning continuous feature values, which allows for faster computation and reduced memory usage. Due to these slight differences, we included both models in the evaluation to determine their performance on the dataset in this study.

To address data imbalance, we performed a comprehensive experimentation by combining the four models with various imbalanced data handling techniques. Specifically, we evaluated the effectiveness of Class weight, Random undersampling [19], and SMOTE [20] in conjunction with the models on our dataset. Our primary goal was to identify the optimal combination that yields the highest performance based on the AUC metric. The models were constructed using the Pipeline methodology in conjunction with GridSearch and Cross Validation techniques to identify the best hyperparameters for each model.

We recognize that the “duration” feature, which represents the outcome of the last inquiry with the customer regarding their application status, may not be available during the initial contact in real-world telemarketing campaigns. To address this, we designed our predictive models in two stages. In the first stage, we developed a model to make predictions without using the “duration” feature. This stage aims to help organizations decide whether to contact a customer based solely on the features available before any interaction. In the second stage, after contact has been initiated and the “duration” feature becomes available through interactions, we built a separate predictive model. This model utilizes the additional information to refine and improve prediction accuracy.

To develop these models, we conducted two sets of experiments. The first set used the full-feature dataset, similar to approaches in prior studies, simulating the second stage where the “duration” feature is available after establishing contact. The second set of experiments excluded the “duration” feature to simulate scenarios where this information is unavailable at the outset. The objective of both experiment sets was to identify the optimal combinations of feature encoding schemes and imbalanced data treatments to maximize the predictive performance of the models. After we obtained the best-performing models in both stages, we applied interpretable ML technique namely SHAP to disclose the underlying rationale behind the model's operation as well as the important features within the dataset. After we obtained the best-performing models in both stages, we applied interpretable ML technique namely SHAP to disclose the underlying rationale behind the model's operation as well as the important features within the dataset.

Interpretable Machine Learning for Model Explanation with SHAP

SHAP was utilized to explain the selected models and determine which features significantly contribute to the prediction. SHAP utilizes Shapley values which are a widely used approach from cooperative game theory to determine the contribution of each feature in the prediction made by a machine learning model. The algorithm calculates feature attributes by determining the difference between when a feature is included and when that feature is excluded as in equation (5).

$$\phi_i = \sum_{S \subseteq F\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\left(x_S\right) \right] \quad (5)$$

where S is a feature subset and F is the set of all features. $f_{S\cup\{i\}}$ indicates the model trained on S when feature i is included, while f_S the model trained on S with i being withheld. As a result, the term $f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\left(x_S\right)$ represents the difference of the prediction between the presence and the absence of feature i. These differences are then weighted based on the number of times each feature appears in different combinations as calculated by the coefficient $\frac{|S|!(|F|-|S|-1)!}{|F|!}$.

Lundberg and Lee [11] show that SHAP is more computation efficient and consistent with human intuition. In case of ensemble tree-based model, the conventional global feature importance for tree-based models including Sabass method, gain, and split count attribution methods are inconsistent

with respect to different tree structures [21]. On the other hand, SHAP gives consistent results on the same dataset regardless of tree structure.

SHAP is a versatile and powerful tool that can be applied to both additive and tree-based models, making it an effective method for model interpretation. It serves as an interpretable tool to uncover and explain the behavior of the chosen model. In addition to its efficiency, SHAP is available as a Python package that offers a wide range of illustrative plots. These plots provide valuable insights at both the model level and the instance level, allowing users to understand how different feature values influence the model's behavior.

The bar plot as shown in Figure 2, is a straightforward diagram that illustrates the importance of features. Each bar represents the absolute contribution of a feature to the predictions. The horizontal line in the plot represents the corresponding SHAP values associated with each feature. In this plot, features with larger SHAP values are considered more important than those with less SHAP values. The length of each bar indicates the magnitude of the feature's contribution to the predictions. By analyzing the bar plot, one can easily identify the features that have the most significant impact on the model's predictions. The relationship between the bar length and the SHAP values provides insights into the relative importance of each feature in influencing the model's behavior.

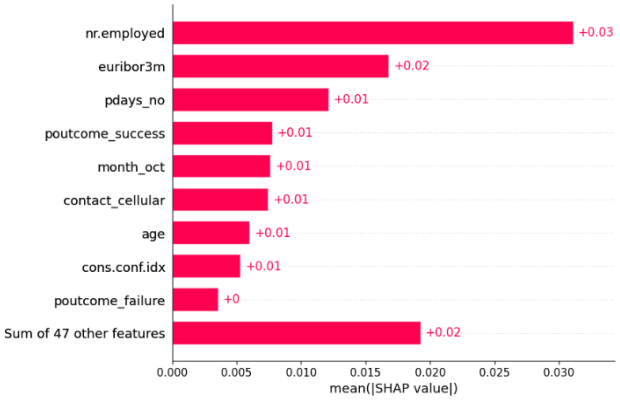

Figure 2 Bar plot ranks feature importance according to the SHAP values.

Another important plot is the Beeswarm plot, as depicted in Figure 3. This plot reflects the importance of each feature, with higher SHAP values indicating a greater likelihood of a positive class prediction. By analyzing the Beeswarm plot, users can identify the features that have the most significant impact on the model's predictions and understand the direction and magnitude of their influence.

The heatmap on the right side of the plot displays the feature values for each instance. High feature values are represented by red circles, while low feature values are denoted by blue circles. The vertical position of each circle along the horizontal line indicates the impact of the corresponding feature on the prediction result. If a circle appears in the section with positive SHAP values, it signifies that the particular feature contributes positively to the prediction. The magnitude of the contribution is indicated by the corresponding SHAP value. This visualization allows for an intuitive understanding of

how different feature values influence the model's predictions, with positive contributions depicted above the horizontal line and their magnitudes indicated by the SHAP values.

Overall, SHAP's effectiveness in interpreting both additive and tree-based models, combined with its powerful visualizations, enables users to gain insights into how feature values impact the model's behavior. This helps improve transparency, trust, and understanding in machine learning models.

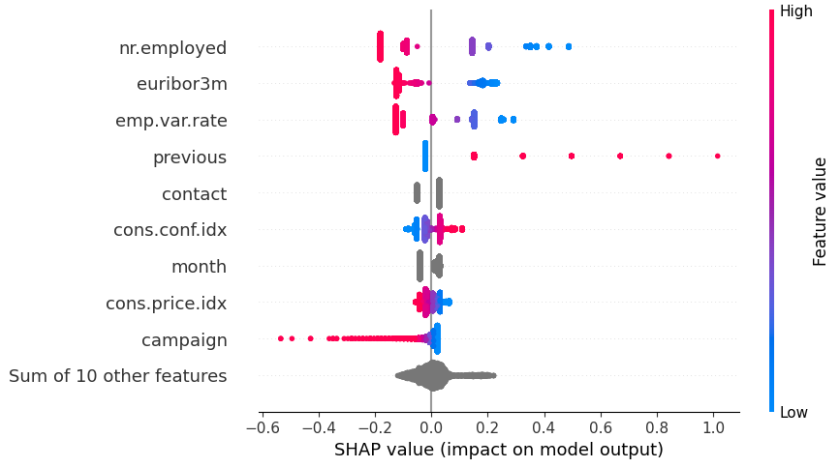

Figure 3. Beeswamp plot shows the relationship between feature values and SHAP values.

Decision plot shows how the model makes the prediction on each instance. It can reveal how each feature is involved in the model's outputs. Figure 4 shows an example of a decision plot of five instances being predicted as negative. Same as Bar and Beeswamp plots, the vertical axis indicates the features. Each line represents the path leading to the prediction of each instance. The horizontal line indicates the decision output of the model. The path which ends at the left side of the horizontal axis indicates that the corresponding instance is predicted as negative sample. Starting from the value 0.00 at the center of the lower horizontal line, the decision path depicts how much each feature on the vertical line contributes to the final decision of the model. Decision plot can be used to investigate and compare the difference between the true predictions and the false predictions effectively.

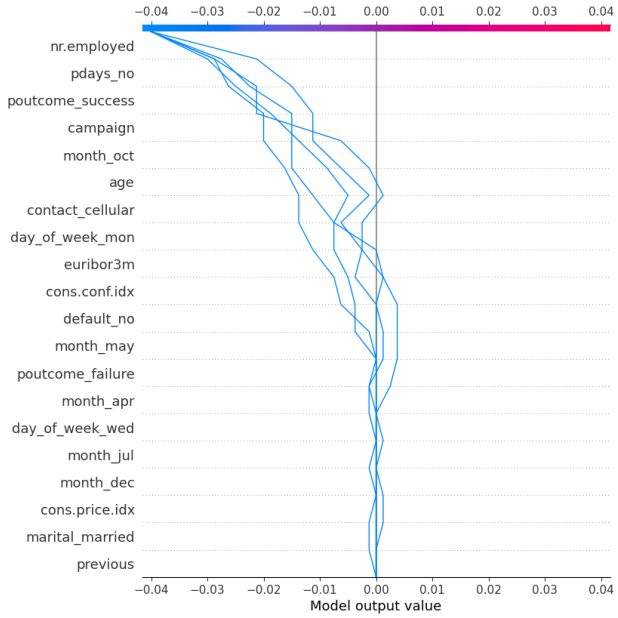

Figure 4 Decision plot shows factors influenced the prediction of each instance by a line.

Results and Discussion

We built two predictive models in which the first model employed all features in the dataset while the second model excludes the "duration" feature. We performed different imbalances handling and feature encoding strategies to find the best combination among different machine learning models. We use AUC scores as the metric for comparing the performance of the models. Next, we use SHAP to investigate the decision mechanism of each model by investigating the features that have more influence on the prediction.

Predictive Models with feature "duration"

The comparative AUC scores of the models in combination with different imbalanced data treatments and numerical encoding approaches are shown in Table 1. We can see that LGBM outperforms other models in all of the combinations. It is important to note that the performance of the models was not significantly affected by the choice of imbalanced data treatments and encoding approaches. As we can see, for the same model, the different combinations result in varying decimal values. This can be attributed to the utilization of grid search, which allowed us to optimize for the best AUC metric across different combinations, thus mitigating the impact of different combinations. We chose the best combination which was LGBM with One-hot encoding and Class weight treatment, with AUC score of 0.948 which was named as LGBM_best1 (Table 2).

Table 2 AUC score of tuning models when including the feature “duration”.

		Model name			
Imbalanced treatments	**Encoding methods**	**LR**	**RF**	**LGBM**	**XGB**
Class weight	OH	0.937	0.913	**0.948**	0.941
	CB	0.934	0.912	**0.947**	0.941
	BN	0.930	0.911	**0.947**	0.941
Under sampling	OH	0.937	0.913	**0.944**	0.941
	CB	0.934	0.914	**0.944**	0.941
	BN	0.930	0.910	**0.943**	0.939
SMOTE	OH	0.937	0.910	**0.946**	0.938
	CB	0.931	0.914	**0.942**	0.936
	BN	0.930	0.911	**0.944**	0.936

In deployment step, for LGBM_best1, we selected an operating threshold to classify instances into the positive ("yes") class. We set the threshold at 0.35 which achieved a recall score of 0.98 for the positive class. The performance of the model is visualized in Figure 5. It is important to note that the accuracy score at this threshold may be lower compared to other studies. However, this decision was driven by our specific application, where the primary objective was to maximize the recruitment of potential customers. Therefore, we prioritized the recall score over the accuracy score when choosing the threshold.

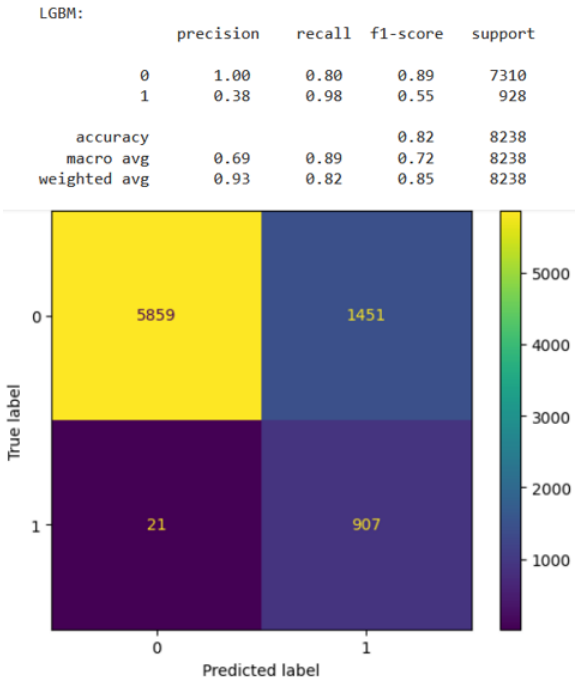

Figure 5 The performance of LGBM_best1 model with the threshold of 0.35.

Predictive Models without the feature "duration"

As with the findings from data including the "duration" feature, the twelve combinations for each model showed minimal differences, as displayed in Table 3. LGBM based models still outperformed the other models. However, in this particular experiment setting, the absence of the "duration" feature resulted in a notable decline in AUC score. The best-performing model achieved an AUC score of 0.95 in the initial set of experiments, whereas in the second set of experiments, the score dropped to approximately 0.8. This decline can be attributed to the significance of the "duration" feature, which strongly correlates with customer decisions. The best combination remained to be LGBM with One-hot encoding and Class weight treatment, with AUC score of 0.805 which was named as LGBM_best2.

In this set of experiments, during deployment step, in order to achieve the recall score as high as 0.9, we adopted ensemble method, in which we aggregate all of the combinations of LGBM based models using equation (6).

$$\hat{y} = \bigvee \left(y_{LGBM_1}, \ldots, y_{LGBM_{12}} \right) \tag{6}$$

where y_{LGBM_i} represents the prediction from the i^{th} combination of LGBM model, $\bigvee$ denotes or operation, and $\hat{y}$ is the ensemble prediction. Note that as we want the model with high recall, we assign class "1" to the instance in which there is at least one LGBM based model predicts as positive class.

Table 3 AUC score of tuning models when excluding the feature "duration".

		Model name			
Imbalanced treatments	**Encoding methods**	**LR**	**RF**	**LGBM**	**XGB**
Class weight	OH	0.794	0.785	**0.805**	0.797
	CB	0.789	0.787	**0.801**	0.795
	BN	0.787	0.786	**0.804**	0.798
Under sampling	OH	0.794	0.785	**0.799**	0.797
	CB	0.789	0.787	**0.793**	0.793
	BN	0.787	0.786	**0.801**	0.797
SMOTE	OH	0.794	0.785	**0.799**	0.789
	CB	0.789	0.787	**0.785**	0.783
	BN	0.787	0.784	**0.796**	0.785

Compared with the results in Figure 5, without feature "duration", we have to contact many more potential customers in order to achieve the same recall score as LGBM_best1. Figure 6 indicates that the model made the positive predictions of 4965+851=5816 people, in which 4965 instances were false positive. This indicates that in order to use this model, we have to contact 5816 people in order to recruit the actual customer of 851 people, compared with the model LGBM_best1 in Figure 5 where there were only 1451 false positive instances. This is the cost that the organization have to pay in the initial contact before feature "duration" can be obtained.

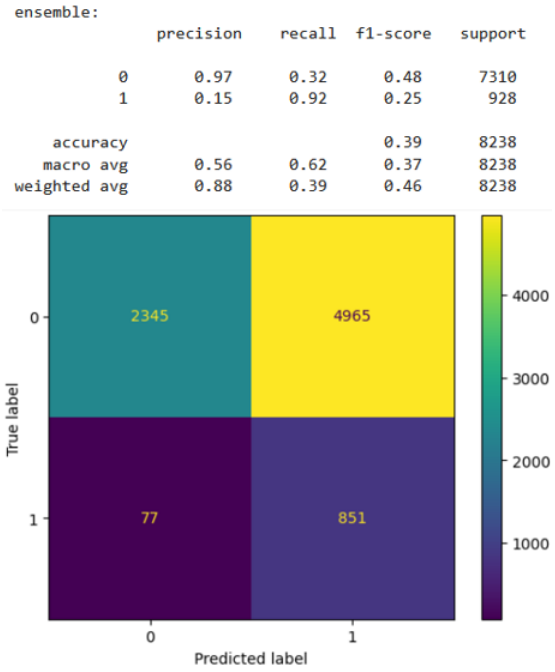

Figure 6 The performance of LGBM_ensemble model.

SHAP explanation for the model using feature "duration"

We first applied SHAP to analyze LGBM_best1 model, trained using features that included the "duration" feature, as shown in Figure 7. As anticipated, the "duration" feature had the highest contribution to the model's predictions. Following that, the 'emp.var.rate' and 'nr.employed' features ranked as the second and third most influential features, respectively. A notable observation from the SHAP plot is that, apart from the "duration" feature, the subsequent four most important features are associated with either the economic environment or the timing of contact. Interestingly, the most significant feature directly related to individual customers, 'age', is the fifth one with its SHAP value roughly one-fourth of the previous feature's SHAP value. This analysis reveals that there is a considerable impact from features connected to the economic climate and the timing of contact. However, individual customer attributes comparatively hold less significance.

Figure 8 depicts Beeswamp plot of the model. It reveals that instances with a low duration (represented by blue dots) have a negative impact on the model, indicating that customers with shorter interaction durations are less likely to accept the campaign. Conversely, customers with longer interaction durations are more likely to accept the campaign. This finding aligns with intuition, as it is expected that customers who are uninterested in the campaign are more likely to decline it immediately.

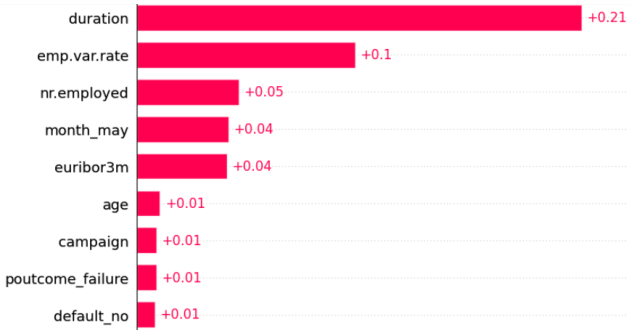

Figure 7 Bar plot shows the SHAP values of each feature from LGBM_best1 model.

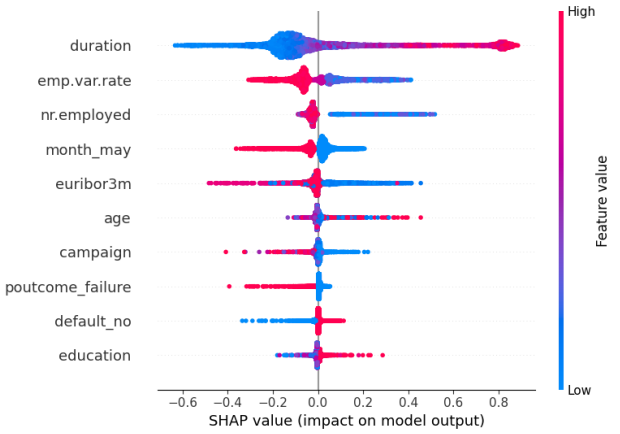

Figure 8 Beeswamp plot shows relationship between feature values and SHAP values from LGBM_best1.

Another notable observation from this plot is the influence of social and economic environment variables on the model's predictions. Specifically, the features 'emp.var.rate' and 'nr.employed', which indicate the employment variation rate and the quarterly average number of employed individuals respectively, exhibit a similar trend where low feature values have a positive impact on the model.

For the 'emp.var.rate' feature, a high value indicates a greater employment variation rate, implying that more people are obtaining employment opportunities [22]. Similarly, a high value for 'nr.employed' indicates a larger number of employed individuals. The results from both variables indicate the same finding that when more people secure employment, there is a tendency for customers to decline the campaign. This finding could be attributed to the fact that the proposed campaign may be related to loans or financial services, and individuals who are already employed may not be interested in such offers.

The 'euribor3m' feature, which represents the Euro Interbank Offered Rate for three months, serves as an indicator of the current market interest rate. Notably, when this feature has a high value, similar to the 'emp.var.rate' and 'nr.employed' features, it exerts a negative impact on positive predictions. This observation can be attributed to the fact that when the interest rate is high, individuals are less inclined to seek loans or similar financial services from banks.

SHAP explanation for the model not using feature "duration"

In this section, SHAP is applied with model LGBM_ensemble, trained without feature "duration". Bar plot of the first eleven important features is shown in Figure 9.

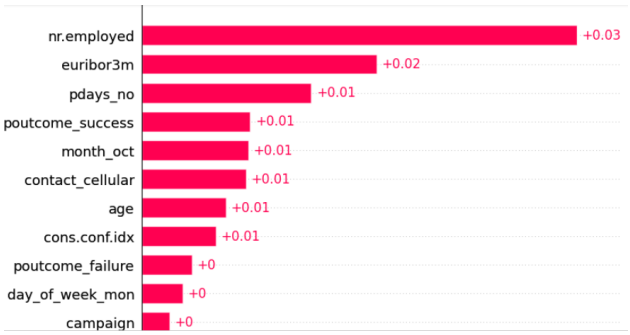

Figure 9 Bar plot shows the SHAP values of each feature from LGBM_ensemble model.

In the absence of the "duration" feature, the significance of the remaining features becomes less pronounced compared to Figure 7. Notably, the two most influential features are 'nr.employed' and 'euribor3m', both of which are related to the economic environment. This finding aligns with the previous model's results, indicating that features related to the economic climate hold greater significance compared to features associated with individual customer characteristics. This discovery is quite remarkable as it contradicts the intuitive assumption that understanding the characteristics of individual customers who are receptive to the campaign is crucial. Instead, the findings suggest that paying closer attention to the economic environment during the initiation of the campaign is of greater importance.

From Figure 10, we observe that the findings regarding the 'nr.employed' feature align with the previous section, where low feature values positively influence the model's predictions. However, in the absence of the "duration" feature, the 'euribor3m' feature does not exhibit a stable pattern, unlike Figure 8. When the feature has a low value, it can have either a positive or negative impact on the predictions. Conversely, when the feature has high values, it does not significantly impact the predictions, as indicated by the SHAP values of the red circles being close to zero.

This observation reflects the fact that, in this model, the 'euribor3m' feature alone cannot be considered in isolation. There is a requirement for interactions between this feature and other features to accurately make predictions. The behavior of the 'euribor3m' feature depends on its relationship with other variables within the model's decision-making process. The remaining features, which are relatively less important, include those related to individual customer characteristics and contact history. For instance, the 'pdays_no' feature indicates whether the customer was contacted in the previous campaign. When this feature has a low value, it suggests that the customer had previous interactions with the bank. Also, the 'poutcome_success' feature, which equals "1" when the previous contact was successful, and "0" otherwise.

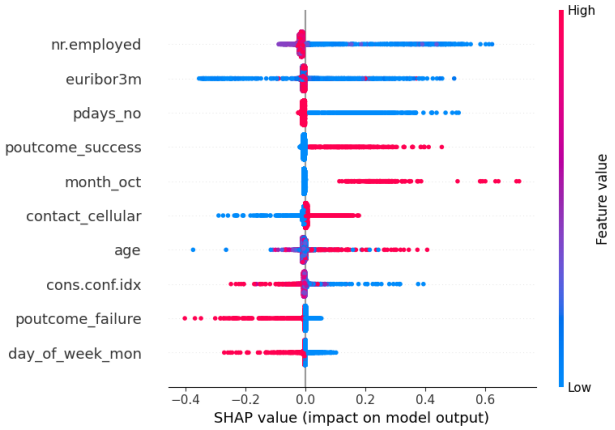

Figure 10 Beeswamp plot shows relationship between feature values and SHAP values from GBM_ensemble.

Notably, the presence of past contact and the success of the previous campaign all have a positive impact on the current predictions. This implies that customers who have been contacted before are more likely to respond positively to the current campaign as well. This finding underscores the importance of considering the history of customer interactions when making predictions and highlights the potential influence of prior engagements on the current outcomes.

As the model without “duration” feature is more useful in real application, we further perform error analysis which focuses on exploring how the model makes incorrect predictions, specifically on false negative samples, which represent potential customers wrongly labeled as negative class by the model's predictions. Figure 11 illustrates the decision plot of false negative samples. As depicted in Figure 11 the negative predictions for these instances are primarily driven by the set of important features, which closely resemble those shown in Figure 10. Notably, the 'nr.employed' feature plays a critical role in causing the incorrect predictions for these false negative samples. The plot illustrates that all features contribute to the negative predictions, suggesting that false negative samples share characteristics highly similar to samples in the negative class.

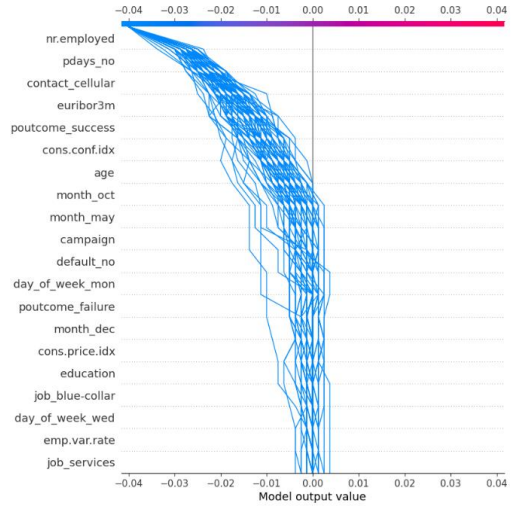

Figure 11 Decision plot from false negative samples.

To gain a comparative view, we also plot the decision plot for both true positive and false negative instances in Figure 12, where true positive instances are represented by red lines, and false negative instances are denoted by purple lines. Figure 12 reveals that the set of important features influencing the predictions for positive instances is akin to those in Figures 9 and 10. The 'nr.employed' feature remains the most significant feature in determining the predictions.

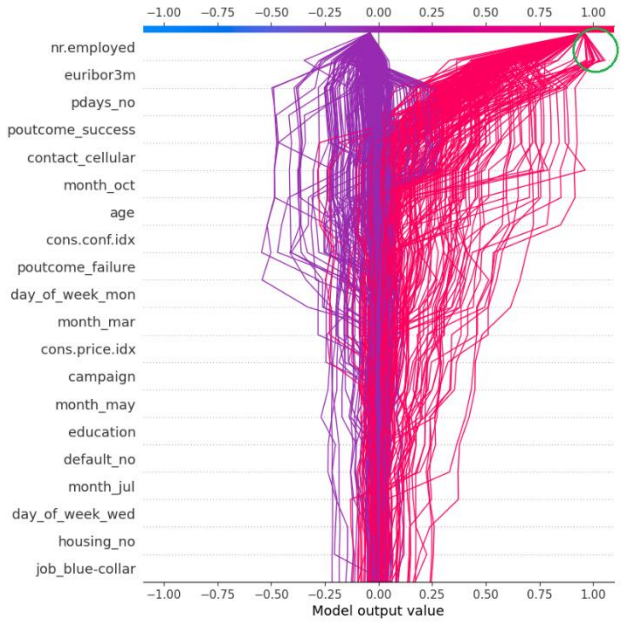

Figure 12 Decision plot from false negative and true positive samples.

To further understand the impact of the 'nr.employed' feature, we create a strip plot in Figure 13 to compare the values of the 'nr.employed' feature between false negative (FN) and true positive (TP) samples. This allows us to analyze the distribution and values of this feature for both types of instances and understand how it contributes to the model's predictions.

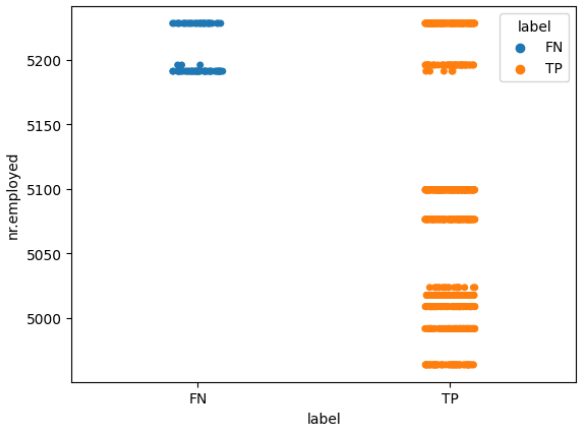

Figure 13 Strip plot shows the values of 'nr.employed' feature for true positive and false negative samples.

Figure 13 shows that for true positive samples, most of the instances have values of 'nr.employed' feature lower than 5150. This finding aligns with the beeswamp plot in Figure 10 that low 'nr.employed' feature contributes to positive predictions. However, note that there are two true positive strips that have 'nr.employed' feature values higher than 5150. These instances correspond with the instances in green circle on the upper right corner as shown in Figure 12 where 'nr.employed' feature gives negative impact. However, overall, these instances are predicted as positive, due to the influence from other features. Based on the error analysis, we have observed that false negative samples exhibit behavior highly similar to the actual negative samples. This similarity suggests that these instances are likely positive samples that lie close to the decision boundary, leading to misclassification as negatives. Adding to this challenge is the highly imbalanced nature of the dataset, with significantly fewer positive samples compared to negative ones. This class imbalance makes it difficult for the model to effectively distinguish false negative samples from the true negative ones.

Conclusions

This study explored the application of Interpretable Machine Learning techniques to analyze data related to telemarketing campaigns for banking products. Two predictive models were developed for distinct deployment phases: before and after customer contact.

Model After Customer Contact: By incorporating all available features, the LightGBM model (LGBM_best1) with One-hot encoding and Class weight treatment achieved the highest AUC score of 0.948. With an operating threshold adjustment to 0.35, the model demonstrated a recall score of 0.98, effectively identifying positive cases.

Model Before Customer Contact: Excluding the “duration” feature to enable immediate model deployment resulted in a lower AUC score of 0.805. However, the ensemble version of LightGBM (LGBM_ensemble) still performed well, achieving a recall score of 0.9, demonstrating its practicality despite the absence of key features.

Feature importance analysis using SHAP revealed that economic factors, such as 'duration,' 'emp.var.rate,' 'nr.employed,' and 'euribor3m,' significantly influenced model predictions. For the LGBM_best1 model, “duration” was the most critical feature, with longer conversation durations positively correlating with product adoption. Conversely, high values of 'nr.employed' and 'euribor3m' negatively impacted predictions, suggesting that individuals with stable employment were less likely to adopt the product. Similar trends were observed in the LGBM_ensemble model, emphasizing the dominant role of economic indicators over individual customer attributes.

Error analysis revealed challenges in distinguishing false negatives from true negatives due to the high class imbalance and the proximity of positive samples to the decision boundary. The 'nr.employed' feature played a key role in these misclassifications, with its higher values often leading to incorrect predictions. This highlights the complexity of addressing imbalanced data in telemarketing applications. While our proposed models demonstrated strong performance on the Bank Marketing dataset, their generalizability to other banks or marketing contexts may be limited. The dataset is based

on historical telemarketing campaigns from a Portuguese bank, and the economic indicators and customer behaviors captured may not reflect those of different regions, time periods, or financial institutions. Additionally, the absence or delay in obtaining certain features (e.g., “duration”) can impact model performance during early-stage deployment. In practical settings, real-time data availability, changes in campaign objectives, and evolving regulatory or privacy constraints may further limit the direct applicability of the model. Threshold settings optimized for recall (e.g., 0.35) may also lead to increased false positives, incurring additional operational costs. These factors should be carefully considered before applying the model in production environments, and periodic retraining with up-to-date and institution-specific data is recommended to maintain accuracy.

Future research could focus on integrating the proposed predictive models into real-world banking workflows, such as CRM systems or automated dialer systems, to dynamically prioritize leads based on updated customer data. Additionally, longitudinal studies could explore how customer responses evolve over time under changing economic conditions or after multiple campaign exposures. From a socio-economic perspective, future work may incorporate external macroeconomic indicators (e.g., unemployment rates, inflation trends) or stratify analysis across customer segments (e.g., income brackets, regions) to better understand how broader societal factors influence campaign responsiveness. Such extensions would enhance the practical relevance and adaptability of predictive models in diverse financial and cultural contexts. Additionally, other interpretable machine learning methods, such as LIME [23], could be investigated for practical use.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Farooqi R, Iqbal N. Performance evaluation for competency of bank telemarketing prediction using data mining techniques. Int J Recent Technol Eng. 2019;8(2):5666-74.
2. Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decis Support Syst. 2014;62:22-31.
3. Koumétio SCK, Cherif W, Silkan H. A data modeling approach for classification problems: application to bank telemarketing prediction. In: Proceedings of the 2nd International Conference on Networking, Information Systems and Security; (NISS 2019); New York: Association for Computing Machinery; 2019. p. 1–7.
4. Koumétio SCK, Hamza T. Improving KNN model for direct marketing prediction in smart cities. In: Machine intelligence and data analytics for sustainable future smart cities. Cham: Springer; 2021. p. 107-18.
5. Koumétio SCK, Gherghina ŞC, Toulni H, Mata P, Mata NN, Martins J. A machine learning framework towards bank telemarketing prediction. J Risk Financ Manag. 2022;15:269.

6. Feng Y, Yin Y, Wang D, Dhamotharan L. A dynamic ensemble selection method for bank telemarketing sales prediction. J Bus Res. 2022;139(4):368-82.
7. Yan C, Li M, Liu W. Prediction of bank telephone marketing results based on improved whale algorithms optimizing S_Kohonen network. Appl Soft Comput. 2020;92.
8. Ghatasheh N, Faris H, AlTaharwa I, Harb Y, Harb A. Business analytics in telemarketing: cost-sensitive analysis of bank campaigns using artificial neural networks. Appl Sci. 2020;10(7):2581.
9. Xie C, Zhang JL, Zhu Y, Xiong BB, Wang GJ. How to improve the success of bank telemarketing? prediction and interpretability analysis based on machine learning. Comput Ind Eng. 2023;175:108874.
10. Molnar C. Interpretable machine learning: a guide for making black box models explainable. 2nd ed. christophm.github.io/interpretable-ml-book; 2022.
11. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Proceedings of the 31st Advances in Neural Information Processing Systems (NeurIPS 2017); 2017 Dec 4–9; Long Beach, CA. Red Hook, NY: Curran Associates Inc.; 2017. p. 4768-77.
12. Hu C, Tan Q, Zhang Q, Li Y, Wang F, Zou X, et al. Application of interpretable machine learning for early prediction of prognosis in acute kidney injury. Comput Struct Biotechnol J. 2022;20:2861-70.
13. Liu Y, Liu Z, Luo X, Zhao H. Diagnosis of Parkinson's disease based on SHAP value feature selection. Biocybern Biomed Eng. 2022;42(3):856-69.
14. Moro S, Rita P, Cortez P. Bank marketing. UCI machine learning repository; 2012.
15. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems (NeurIPS 2018); 2018 Dec 3–8; Montréal, Canada. Curran Associates Inc.; 2018. p. 6639–49.
16. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13-17, San Francisco: Association for Computing Machinery; 2016. p. 785-94.
17. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA. Red Hook, NY: Curran Associates Inc.; 2017. p. 3149-57.
18. Borisov V, Leemann T, Seßler K, Haug J, Pawelczyk M, Kasneci G. Deep neural networks and tabular data: a survey. IEEE Trans Neural Netw Learn Syst. 2022;1:1-21.
19. Hasanin T, Khoshgoftaar TM. The effects of random undersampling with simulated class imbalance for big data. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI); 2018 July 6-9; Salt Lake City, UT: IEEE Press; 2018. p. 70-9.
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321-57.

21. Lundberg SM, Erion GG, Lee SI. Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888; 2018.
22. Miller M, Pavosevich R. Alternative methods of experience rating unemployment insurance employer taxes. Public Budg Finance. 2019;39(4):28-47.
23. Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13-17, San Francisco: Association for Computing Machinery; 2016. p. 1135-44.