sustainability

MDPI

Article

Predictive Churn Modeling for Sustainable Business in the Telecommunication Industry: Optimized Weighted Ensemble Machine Learning

Wee How Khoh [1], Ying Han Pang [1,*], Shih Yin Ooi [1], Lillian-Yee-Kiaw Wang [2] and Quan Wei Poh [3]

[1] Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
[2] School of Information Technology, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, Subang Jaya 47500, Malaysia
[3] Winnefy Enterprise, Jalan SD 2/6, Taman Sri Duyong 2, Melaka 75460, Malaysia
* Correspondence: yhpang@mmu.edu.my

Abstract: Customers are prominent resources in every business for its sustainability. Therefore, predicting customer churn is significant for reducing churn, particularly in the high-churn-rate telecommunications business. To identify customers at risk of churning, tactical marketing actions can be strategized to raise the likelihood of the churn-probable customers remaining as customers. This might provide a corporation with significant savings. Hence, in this work, a churn prediction system is developed to assist telecommunication operators in detecting potential churn customers. In the proposed framework, the input data quality is improved through the processes of exploratory data analysis and data preprocessing for identifying data errors and comprehending data patterns. Then, feature engineering and data sampling processes are performed to transform the captured data into an appropriate form for classification and imbalanced data handling. An optimized ensemble learning model is proposed for classification in this framework. Unlike other ensemble models, the proposed classification model is an optimized weighted soft voting ensemble with a sequence of weights applied to weigh the prediction of each base learner with the hypothesis that specific base learners in the ensemble have more skill than others. In this optimization, Powell's optimization algorithm is applied to optimize the ensemble weights of influence according to the base learners' importance. The efficiency of the proposed optimally weighted ensemble learning model is evaluated in a real-world database. The empirical results show that the proposed customer churn prediction system achieves a promising performance with an accuracy score of 84% and an F1 score of 83.42%. Existing customer churn prediction systems are studied. We achieved a higher prediction accuracy than the other systems, including machine learning and deep learning models.

Keywords: churn prediction; business sustainability; telecommunication; ensemble learning; weight optimization

Citation: Khoh, W.H.; Pang, Y.H.; Ooi, S.Y.; Wang, L.-Y.-K.; Poh, Q.W. Predictive Churn Modeling for Sustainable Business in the Telecommunication Industry: Optimized Weighted Ensemble Machine Learning. *Sustainability* **2023**, *15*, 8631. https://doi.org/10.3390/su15118631

Academic Editor: Manuel Fernandez-Veiga

Received: 16 February 2023
Revised: 11 May 2023
Accepted: 14 May 2023
Published: 25 May 2023

1. Introduction

Customers are prominent resources in every business for its sustainability. It has been claimed that acquiring a new customer is more costly than retaining an existing customer due to advertisement costs, concession costs, workforce costs, and other costs [1–3]. Customer churn is a crucial concern in the rapidly growing and competitive telecommunication industry, while customer loyalty is the key to business sustainability and profitability. Existing customers often express dissatisfaction with the network connectivity, spam marketing emails, internet speed, and complicated billing. Consequently, it is no wonder that the telecommunications industry has a high customer churn rate. Statistics show that the average churn rate in telco businesses is 22%, in contrast with 19% in IT and 16% in professional services [4]. The churn rate, or customer attrition rate, is the rate of customers who discontinue using a company's product or service for a specific duration. It can be expressed

as the percentage of telco service subscribers who stop their subscriptions within a given time frame. Reducing churn is more crucial than ever, especially in the telecommunication industry and present economic strife. Each time a customer leaves signifies a substantial investment loss. Hence, customer churn prediction is significant. Being able to identify customers at risk of churning allows tactical marketing actions to be strategized to raise the likelihood of the churn-probable customers remaining as customers. This could proffer significant savings to a business.

Therefore, customer churn prediction is a popular data science research topic in the commercial sector. There are a plethora of machine learning classifiers that have been proposed to analyze customer data for predicting customer churn. These include single classifiers, such as support vector machines, naïve Bayes, decision trees, logistic regression, and k-nearest neighbors, and ensemble classifiers, such as AdaBoost, gradient boosting, XGBoost, CatBoost, and random forests [1,5–12]. It has been asserted that ensemble classifiers perform better than single classifiers [13]. There are two types of ensemble classifiers: homogeneous and heterogeneous. Homogeneous ensemble classifiers utilize the same base classifier but adopt different sampling methods. In contrast, heterogeneous ensemble classifiers combine multiple base classifiers with diverse characteristics, which can be single classifiers and/or homogeneous ensemble classifiers. Due to the simple algorithm implementation and computational efficiency, homogeneous ensemble classifiers have been widely employed in various applications, including churn prediction. However, their performance may be limited in capturing diverse aspects of real-world data. Customer churn prediction is a complex task that involves working with large and complex real-world customer data. Hence, heterogeneous ensemble classifiers are considered more appropriate for customer churn prediction because they leverage several base classifiers' strengths, allowing them to capture diverse aspects of the data.

This work proposes a heterogeneous ensemble for customer churn prediction in the telecommunication industry. Exploratory data analysis and data preprocessing are performed in the proposed churn prediction model to identify errors and outliers and understand data patterns for data quality improvement. Next, feature engineering and data sampling are employed to transform the captured data into an appropriate form for machine learning and addressing imbalanced data. Lastly, an ensemble learning model is built to learn the processed data and predict churn. This work considers different base learners with different characteristics in the ensemble to increase diversity. The predictions of these base learners are aggregated using weighted soft voting. In the proposed model, Powell's optimization algorithm determines and optimizes the ensemble weights of influence according to the base learners' importance. A sequence of weights is computed to weigh the prediction of each base learner with the hypothesis that specific base learners in the ensemble have more skill than the others and contribute more to making predictions. The efficiency of the proposed optimally weighted ensemble learning model is evaluated in a publicly available database, i.e., Cell2Cell. The experimental analysis reveals the proposed enhanced ensemble learner's superior performance compared with other ensemble and deep learning models.

2. Related Work

Various methods have been proposed to predict customer churn in the telecommunication industry. In the literature, most churn prediction works focus on machine learning models. For instance, Huang et al. applied seven prediction models for land-line customer churn prediction [7]. The models include linear classification, logistic regression, C4.5 decision trees, naïve Bayes, support vector machines, evolutionary data mining algorithms, and multilayer perceptron neural networks. Furthermore, Jain et al. presented churn prediction using logistic regression and LogitBoost in a real-world database, i.e., Orange, an American telecom company dataset [14]. Both the logistic regression and LogitBoost model demonstrated encouraging prediction performance. Ullah et al. presented a churn prediction model using a random forest [2]. In the proposed model, information gain and

correlation attribute ranking filters were implemented for feature selection. After that, the selected significant features were input into a random forest for churn classification. The reported empirical results indicated the superiority of the proposed model using a random forest classifier for customer churn prediction.

In addition, Lalwani et al. presented a churn prediction system for telecom data [15]. This work conducted data preprocessing and feature analysis in the first two phases. Next, feature selection using a gravitational search algorithm was performed, followed by training and testing data preparation. In this proposed system, multiple individual and ensemble learning models were applied to evaluate the performance of the models. The examined classifiers include decision trees, naïve Bayes, logistic regression, random forests, support vector machines, the AdaBoost classifier, the CatBoost classifier, the XGBoost classifier, etc. The reported experimental results demonstrated that the ensemble learning techniques (i.e., the AdaBoost classifier and XGBoost classifier) provide optimal accuracy with an area-under-curve (AUC) score of 84% in the churn prediction task. The work in [5] manages telecom customer data and proposes a churn prediction model based on machine learning algorithms. Prior data prediction, preparation, and cleaning are performed to prepare quality data for machine learning. A dataset with approximately 3300 instances was used for model evaluation. The decision tree classifier demonstrated an accuracy of 98%, whereas the classification model developed from logistic regression achieved an accuracy of 80%. Abhinav and Vijay also explored several machine learning algorithms for customer churn prediction in the Telcom industry [11]. They employed a decision tree, a random forest, and XGBoost in the proposed classification model.

Deep learning approaches have become increasingly popular in recent years due to their exceptional performance [16]. The algorithms can automatically learn hierarchical data representations and extract progressively abstract and complex features. Various deep learning approaches have been proposed for churn prediction, and promising performance results have been reported [17–21]. For instance, Umayaparvathi and Iyakutti explored three deep learning models in the CrowdAnalytix and Cell2Cell datasets for churn prediction [22]. The models are (1) a small feedforward neural network with three layers (one input layer and two dense layers), (2) a large feedforward neural network with four layers (one input layer and three dense layers), and (3) a convolutional neural network. The experimental results exhibit that the deep learning models are on par with popular machine learning classifiers such as random forests and support vector machines. In addition, Ahmed et al. constructed a transfer learning model by tuning several pre-trained convolutional neural networks [23]. This work first transforms telecom data into a two-dimensional image format. Next, the transformed data are further processed for feature analysis. In this architecture, pre-trained convolutional neural networks are used as base classifiers, and genetic programming and AdaBoost are used as meta-classifiers. The authors reported that their proposed model could obtain an accuracy of 75.4% and 68.2% in the Orange and Cell2Cell databases, respectively.

Samah et al. constructed a deep backpropagation artificial neural network called Deep-BP-ANN using the variance thresholding and lasso regression feature selection methods [24]. The efficacy of the proposed model was assessed using two datasets: the IBM Telco and Cell2Cell databases. The empirical results show the promising churn prediction performance of the proposed Deep-BP-ANN model. Deep learning provides powerful applications to businesses. Specifically, deep learning can increase customer churn prediction accuracy and efficacy. However, applying deep learning to business use cases creates some challenges. These models are intricate and expensive to design and deploy. The deep learning algorithm/architecture has to be tailored and carefully tuned to the particular use case or database that they are applied to. Else, suboptimal performance is obtained. This is because different databases may have distinctive properties, such as data size, distribution, and quality, which may affect the performance of the deep learning algorithm. On top of this, deep learning is more computationally intensive, requiring high computational resources due to the model's gigantic hyperparameters.

These hyperparameters must be optimized during model training, and this optimization necessitates extensive computation, which is very resource-intensive.

3. System Design

This work proposes an enhanced ensemble machine learning approach for customer churn prediction. The proposed system design framework is illustrated in Figure 1. Firstly, exploratory data analysis (EDA) and data preprocessing are performed. This process is essential to detect incorrect or missing values, noise, and inconsistencies and extract meaningful insights from the data. Next, feature engineering is performed to transform the raw data into an appropriate form for machine learning. One of the significant challenges when dealing with customer churn analysis is the imbalanced data. Data-class-imbalanced datasets are standard in the telecommunication industry, i.e., churners are in the minority [8]. Thus, imbalanced dataset handling is vital to avoid hampering a model's accuracy. In other words, if a dataset is imbalanced, the model shows relatively high accuracy by predicting the majority class but fails to capture the minority class. This would be a problem for customer churn prediction since identifying churners, i.e., the minority class, is the main objective. Lastly, model generation is performed to build a reliable classification model. The built model is tested with unknown test data for performance assessment.

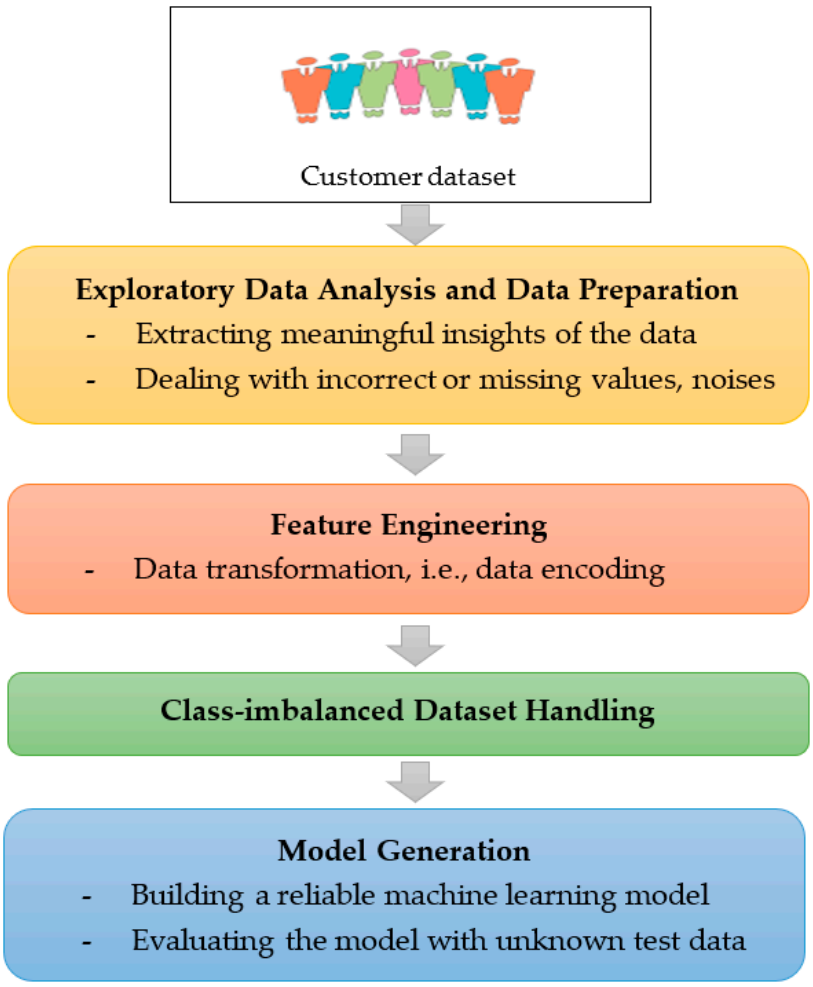


Figure 1. System design framework for churn prediction.

4. System Implementation

4.1. Dataset

In this study, a publicly available telecommunication dataset, Cell2Cell, is adopted to assess the proposed optimized weighted ensemble learner. It consists of open-source data

collected by the Teradata Center at Duke University. The dataset can be downloaded from https://www.kaggle.com/datasets/jpacse/datasets-for-churn-telecom (accessed on 13 May 2023). This is a real dataset collected from customers of the Cell2Cell telecom company with six information categories of features:

- Customer demographics and personal data;
- Customer care service data;
- Customer credit scores;
- Billing and payment data;
- Customer usage patterns;
- Value-adding services.

4.2. Exploratory Data Analysis and Data Preprocessing

Before data modeling, comprehending data variables and structures is a preliminary yet crucial process in a customer churn prediction system. The raw unprocessed dataset will likely contain redundancies, i.e., outliers, missing values, and irregular or non-essential data. This may cause the model to overfit or underfit during training, negatively impacting the model's performance on new data. Hence, exploratory data analysis, coined as EDA, should be performed to comprehend different aspects of the data to detect any data anomalies, "clean", and preprocess the data before modeling the data. Table 1 records a description of the Cell2Cell dataset.

Table 1. Description of the Cell2Cell dataset.

Numerical Feature	Data Format	Categorical Feature	Data Format
CustomerID	Int	Churn	(Yes/No)
MonthlyRevenue	float	ServiceArea	string
MonthlyMinutes	float	ChildrenInHH	(Yes/No)
TotalRecurringCharge	float	HandsetRefurbished	(Yes/No)
DirectorAssistedCalls	float	HandsetWebCapable	(Yes/No)
OverageMinutes	float	TruckOwner	(Yes/No)
RoamingCalls	float	RVOwner	(Yes/No)
PercChangeMinutes	float	Homeownership	(Known/Unknown)
PercChangeRevenues	float	BuysViaMailOrder	(Yes/No)
DroppedCalls	float	RespondsToMailOffers	(Yes/No)
BlockedCalls	float	OptOutMailings	(Yes/No)
UnansweredCalls	float	NonUSTravel	(Yes/No)
CustomerCareCalls	float	OwnsComputer	(Yes/No)
ThreewayCalls	float	HasCreditCard	(Yes/No)
ReceivedCalls	float	NewCellphoneUser	(Yes/No)
OutboundCalls	float	NotNewCellphoneUser	(Yes/No)
InboundCalls	float	OwnsMotorcycle	(Yes/No)
PeakCallsInOut	float	HandsetPrice	string
OffPeakCallsInOut	float	MadeCallToRetentionTeam	(Yes/No)
DroppedBlockedCalls	float	CreditRating	string
CallForwardingCalls	float	PrizmCode	(Other/Suburban/Town/Rural)
CallWaitingCalls	float	Occupation	(Other/Professional/Crafts/Clerical/Self/Retired/Student/Homemaker)
MonthsInService	int	MaritalStatus	(Unknown/Yes/No)
UniqueSubs	int		
ActiveSubs	int		
Handsets	float		
HandsetModels	float		
CurrentEquipmentDays	float		
AgeHH1	float		
AgeHH2	float		

Table 1. *Cont.*

Numerical Feature	Data Format	Categorical Feature	Data Format
RetentionCalls	int		
RetentionOffersAccepted	int		
ReferralsMadeBySubscriber	int		
IncomeGroup	int		
AdjustmentsToCreditRating	int		

Some feature variables have missing values, such as MonthlyRevenue, MonthlyMinutes, TotalRecurringCharge, DirectorAssistedCalls, OverageMinutes, RoamingCalls, PercChangeMinutes, PercChangeRevenues, ServiceArea, Handsets, HandsetModels, CurrentEquipmentDays, AgeHH1, and AgeHH2, as shown in Table 2. These missing values are replaced with zeros. Besides that, the variables CustomerID and ServiceArea, which are not relevant to churn prediction, are discarded [22].

Table 2. The number of missing values in the feature variables.

Feature Variable	Number of Missing Values
MonthlyRevenue	156
MonthlyMinutes	156
TotalRecurringCharge	156
DirectorAssistedCalls	156
OverageMinutes	156
RoamingCalls	156
PercChangeMinutes	367
PercChangeRevenues	367
ServiceArea	24
Handsets	1
HandsetModels	1
CurrentEquipmentDays	1
AgeHH1	909
AgeHH2	909

4.3. Feature Engineering

Most machine learning models are not able to deal with categorical variables. Hence, those categorical data are encoded into numerical form before using them to fit and evaluate a model. This study transforms categorical values using label encoding into a numeric value between 0 and the number of classes minus 1. For instance, the categorical variable MaritalStatus contains three distinct classes (i.e., *No*, *Unknown*, and *Yes*), and the converted values are (0, 1, and 2), and the categorical variable PrizmCode contains four distinct classes (i.e., *Other*, *Rural*, *Suburban*, and *Town*), and the converted values are (0, 1, 2, and 3), as shown in Figures 2 and 3, respectively.

4.4. Class-Imbalanced Dataset Handling

A class-imbalanced scenario usually occurs when solving real-world classification tasks, especially in binary classification. Customer churn prediction is often an imbalanced-class binary classification wherein the majority class (negative/non-churn class) is significantly larger than the minority class (positive/churn class). Similar to the Cell2Cell dataset, the data are seriously imbalanced, with 71.2% being non-churn and 28.8% churn, as seen in Figure 4. In the figure, 0.0 denotes the non-churn class, and 1.0 indicates the churn class. We can see in the figure that there are much more non-churn data samples (yellow region) than churn samples (orange area).

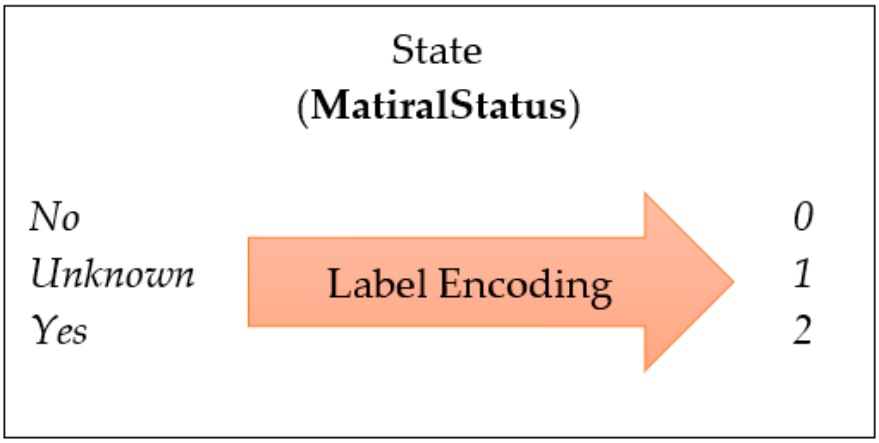


Figure 2. Sample of label encoding for the categorical variable MaritalStatus.

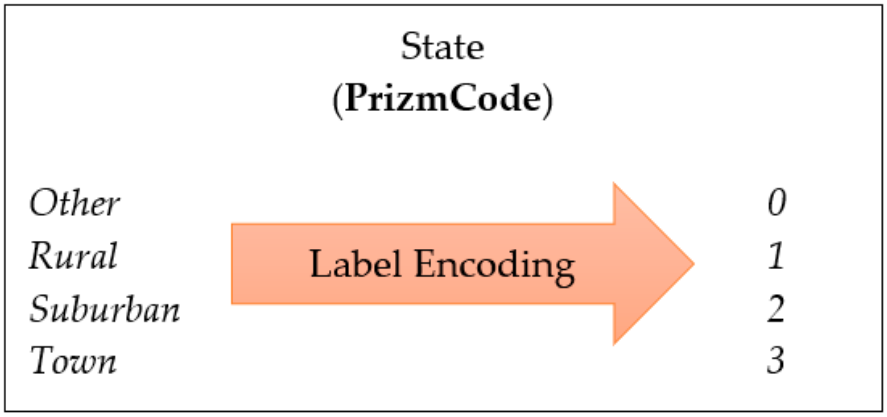


Figure 3. Sample of label encoding for the categorical variable PrizmCode.

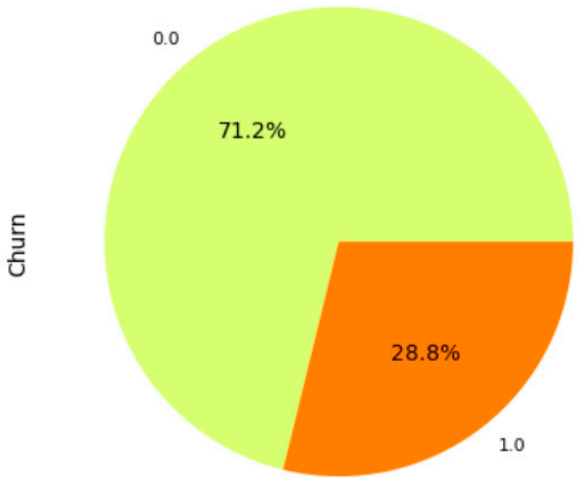


Figure 4. Percentages of non-churn (denoted as 0.0) and churn (marked as 1.0) in the original Cell2Cell dataset.

In our study, we examine the machine learning models under three scenarios:

- Do nothing (dealing with the original imbalanced dataset);
- Classification with the undersampling technique (deleting observations from the majority class to balance the data);
- Classification with the oversampling method (creating synthetic samples of the minority class by employing the synthetic minority over-sampling technique (SMOTE).

4.5. Model Generation: Optimally Weighted Ensemble Learner

It is supposed that amalgamating the predictive information of multiple machine learning models in a given classification task could return a better performance than an individual algorithm could. In other words, uniting various machine learning models into a meta-classifier could effectively resolve the poor generalization potential observed in the respective model. This meta-approach is known as ensemble learning. This paper proposes a heterogeneous ensemble for assisting telecommunication operators in apprehending customer attributes and then computing the susceptibility that a customer will stop using and paying for the products or services. Different types of base learners with different characteristics are considered in the ensemble to leverage the strengths of these classifiers and capture diverse aspects of the data. Powell's optimization algorithm determines and optimizes the ensemble weights according to the base learners' importance.

Figure 5 illustrates the proposed optimized weighted ensemble learner for predicting customer churn. In the figure, the input data are preprocessed and engineered for better data quality before performing feature analysis and classification. This is crucial in machine learning and can significantly impact the performance of a classification model. The proposed ensemble learning framework comprises two blocks: First, the base learners of the *k*-nearest neighbors (KNN), CatBoost, and random forest algorithms analyze the engineered customer features computed from the previous step on an individual basis and provide the prediction results based on personal decisions. According to the prediction performance of the base learners, Powell's optimization algorithm is applied to determine the optimal weights of influence. Specifically, the individual prediction performances of each base classifier are mingled with a soft voting ensemble learner according to the computed optimized weights to distinguish between churners and non-churners. The optimized weights are calculated to allocate smaller weights to weaker base learners and larger weights to stronger base learners.

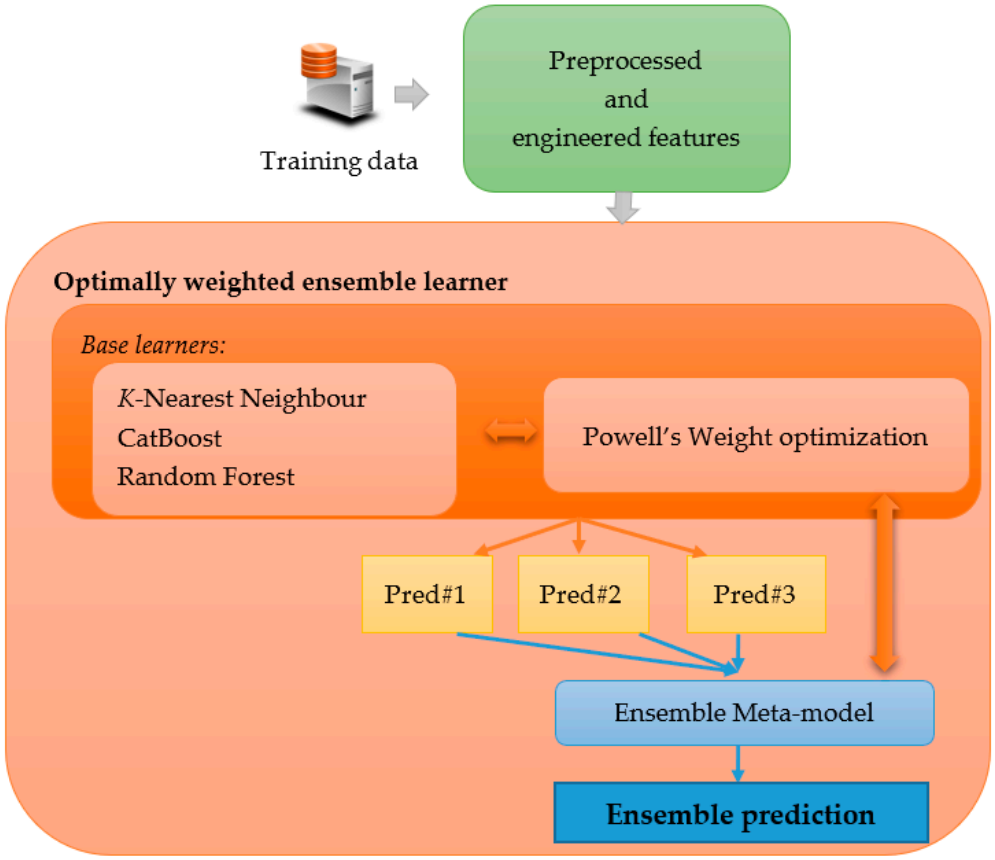


Figure 5. The proposed optimized weighted ensemble learner.

4.5.1. K-Nearest Neighbors (KNN)

KNN is a relatively simple classifier without objective functions or parameters. This algorithm analyses the correlation between test and training samples. A test sample is classified by computing the distances between *k*-nearest samples and the test sample. In this study, the metric used for distance computation is Minkowski distance:

$$dist = \left(\sum_{j=1}^{n} |x_j - z_j|^r \right)^{1/r} \quad (1)$$

where x_j and z_j are the coordinates of sample points in a multi-dimensional space; *dist* is the Manhattan distance when $r = 1$ and represents the Euclidean distance when $r = 2$. A *k*-dimensional tree (i.e., a ball tree) is established as a diagnostic model to minimize the redundancy of the nearest neighbor searching and speed up the diagnosis time.

4.5.2. CatBoost

CatBoost is an algorithm for gradient boosting on decision trees by greedily constructing combinations. CatBoost is becoming popular due to its efficiency features such as fast GPU training, ease of use, and working well with categorical variables [25]. This algorithm has been explored in churn prediction in different sectors, and encouraging performances have been obtained [6,12,26,27]. Let data with samples $T = \{(X_j, y_j)\}_{j=1,\ldots,m}$, and $X_j = \left(x_j^1, \ldots, x_j^n\right)$ is a vector of n features and response feature $y_j \in \mathbb{R}$, which is binary or encoded as a numerical feature. (X_j, y_j) are independently and identically distributed according to some unknown distribution $P(.,.)$. The learning task of Catboost is to train a function $F : \mathbb{R}^n \to \mathbb{R}$ by minimizing the expected loss, as shown below [28]:

$$\mathcal{L}(F) := \mathbb{E}L(y, F(X)) \quad (2)$$

where $L(.,.)$ is a smooth loss function, and (X,y) is testing data sampled from the training data T.

Gradient boosting iteratively constructs a sequence of approximations $F^t : \mathbb{R}^m \to \mathbb{R}, t = 0, 1, \ldots$ in a greedy fashion [29]. From the previous approximation F^{t-1}, F^t is computed in an additive process, i.e., $F^t = F^{t-1} + \alpha g^t$ with a step size α, and function $g^t : \mathbb{R}^n \to \mathbb{R}$, which is a base predictor, is selected from a set of functions G in order to minimize the expected loss, such that

$$g^t = \arg\min_{g \in G} \mathcal{L}\left(F^{t-1} + g\right) = \arg\min_{g \in G} \mathbb{E}L\left(y, F^{t-1}(X) + g(X)\right) \quad (3)$$

CatBoost is an enhanced gradient boosting decision tree algorithm [30]. It incorporates the features detailed below:

- Optimal split selection: The algorithm in CatBoost is designed to choose the best-split point for each node in a decision tree by minimizing the loss function concerning the split ends. This is accomplished by constructing a function that approximates the loss function as a function of the split point. Then, a Newton–Raphson solver is applied to seek the minimum of the procedure. This optimal split selection algorithm allows faster convergence during training and improved accuracy.
- Reduced overfitting feature: Overfitting is a common issue in gradient boosting, especially when the dataset is small or noisy. CatBoost incorporates several features for preventing overfitting. One of them is ordered boosting, a novel gradient-based regularization technique that penalizes complex models that overfit the data. Furthermore, using a per-iteration learning rate enables the model to adapt to the problem's complexity at each iteration, preventing overfitting.

4.5.3. Random Forest

A random forest is a versatile machine learning method that grows and links many decision trees to create a forest. In other words, this supervised classifier constructs a series of weak decision trees from a random sample of the training dataset. It reiterates the process with many random samples and makes a final decision based on majority voting. Algorithm 1 presents the algorithm of a random forest.

Algorithm 1: The algorithm of random forest

Let T be a training dataset, $T = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
Let $b = b_1(x), b_2(x), \ldots, b_h(x)$*, an ensemble of weak classifiers*
Each b_i *is a decision tree and the parameters of the tree are defined as* $\Phi_i = \left(\varphi_{i1}, \varphi_{i2}, \ldots, \varphi_{iq}\right)$
Each decision tree i leads to a classifier $b_i(X) = b(X \mid \Phi_i)$
Final Prediction = high voted predicted target based on b

4.5.4. Weighted Soft Voting Ensemble Learner with Optimized Ensemble Weights Using Powell's Optimization Algorithm

This study proposes a weight-optimized ensemble learner for predicting customer churn based on weighted soft voting. An ensemble learner is a meta-classifier that considers several base learners' predictions for the sake of producing better predictive performance through majority voting. There are two types of voting techniques: hard voting and soft voting. In the former, the final prediction is concluded with the highest number of votes. On the other hand, the latter entails merging the probabilities of each base learner's prediction and opting for the prediction with the highest total likelihood. The soft voting strategy is adopted in this study since it presents a better performance than hard voting [31,32].

In soft voting, the predictions of each base learner, p, are weighted according to the classifier's significance and summed up. Next, the target/class label with the most significant sum of the weighted probabilities wins the vote, such that

$$\hat{y} = \arg\max_{i} \sum_{j}^{H} w_j p_{ij} \tag{4}$$

where w_j is the weight of influence allocated to the jth learner.

In this proposed customer churn prediction ensemble learner, Powell's optimization algorithm is adopted to determine the ensemble weights of the soft voting ensemble learner. This is because the algorithm is a derivative-free and efficient optimization algorithm that can handle nonlinear models. Customer churn data are typically nonlinear data. The relationship between input variables (i.e., data attributes) and a target variable (customer churn) is often nonlinear and complex. Powell's optimization algorithm does not require the computation of gradient information, making it suitable for optimizing models in customer churn prediction. In the ensemble weight computation, smaller weights are allocated to weaker base learners, while larger weights are assigned to stronger base learners. Powell's optimization is a single-shot algorithm for searching a local minimum of a function. This algorithm minimizes the function through a bi-directional search by shifting along one direction until a minimum is attained. From there, it moves along the next direction until a minimum point is accessed, and so on, cycling through the whole set of directions until the fit statistic is minimized for a particular iteration. The algorithm proceeds via iterations until no significant improvement is produced [33]. Algorithm 2 presents Powell's optimization algorithm [34].

Next, the individual prediction results of each base classifier are mingled with a soft voting ensemble learner according to the optimized weights to distinguish between churners and non-churners. The target/class label (i.e., churn or non-churn) with the largest sum of weighted probabilities wins the vote.

Algorithm 2: Powell's optimization algorithm

Initialize the starting point x_1, *independent vectors* $d_i = e_i (i = 1, \ldots, D)$, *the tolerance for stopping criteria* ε, *set* $f(1) = f(x_1), x_c = x_1, \dot{K} = 1$
while *(stopping criterion is not met, i.e.* $|\Delta f| > \varepsilon$*)* ***do***
  for *i = 1 to D* ***do***
    if $(\dot{K} >= 2)$ ***then***
      $d_i = d_{i+1}$
    endif
    $x_{i+1} = x_i + \lambda_i d_i$, λ_i *is determined by minimizing* $f(x_{i+1})$
  endfor
  $d_{i+1} = \sum_1^D \lambda_i * d_i = x_{D+1} - x_D, \quad x_c(k+1) = x_{D+1} + \lambda_k d_{i+1},$
  $f(k+1) = f(x_c(k+1)),$
  $k = k+1, \qquad x_1 = x_c(k), \qquad \Delta f = f(k) - f(k-1)$
endwhile

5. Experimental Results and Discussion

5.1. Experimental Setup and Performance Metrics

This experiment uses Jupyter Notebook with Python 3 with an Intel Core i7 processor with 4.20 GHz and 48 GB of RAM. The dataset used is the telecommunication dataset, Cell2Cell, which has been explained in Section 4. In this study, three database versions are evaluated: the original version, i.e., the imbalanced version; the downsampling version to rectify the imbalanced data; and the oversampling version for imbalanced data rectification. Both sampling processes are applied only to the training set. There are no changes to the testing set. In the down-sampling, the count of training samples in the majority class is reduced.

On the other hand, the over-sampling process injects the synthetically generated minority class's data points into the dataset so that the counts of both majority and minority classes are almost the same. This helps prevent the model from inclining towards the majority class. In this study, SMOTE (synthetic minority oversampling technique) is adopted for oversampling. The algorithm performs based on the *k*-nearest neighbors algorithm by synthetically creating data points for the minority class, whereby

$$x' = x + rand\ (0,1) \times |x - x_k| \tag{5}$$

where x' represents the synthetic sample, x is the original sample, $rand\ (0,1)$ is a random number in the range between 0 and 1, and x_k is the *k*-nearest neighbor. The best-performing version will be adopted for the subsequent experiments.

On top of the accuracy score, different performance metrics are used to assess the classifiers. This is mainly because the accuracy metric is inappropriate for imbalanced data, especially in binary classification. In the customer churn database, the churned customer classes are considerably fewer compared with the non-churned customer classes. A high accuracy is achieved with a non-skilled classifier that only predicts the majority class. Hence, other performance metrics such as precision, recall, F1, and area under the curve (AUC) are used in this study. The following equations present the formulations of the performance metrics for our churn prediction tasks.

$$accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{6}$$

$$precision = \frac{T_p}{T_p + F_p} \tag{7}$$

$$recall = \frac{T_p}{T_p + F_n} \tag{8}$$

$$F1 = 2 \times \frac{Precision \times Recall}{recision + Recall} \tag{9}$$

$$AUC = \int_{-\infty}^{\infty} y(t)dx(t) \tag{10}$$

where T_p is a true positive, T_n is a true negative, F_p is a false positive, and F_n is a false negative.

5.2. Results and Discussion

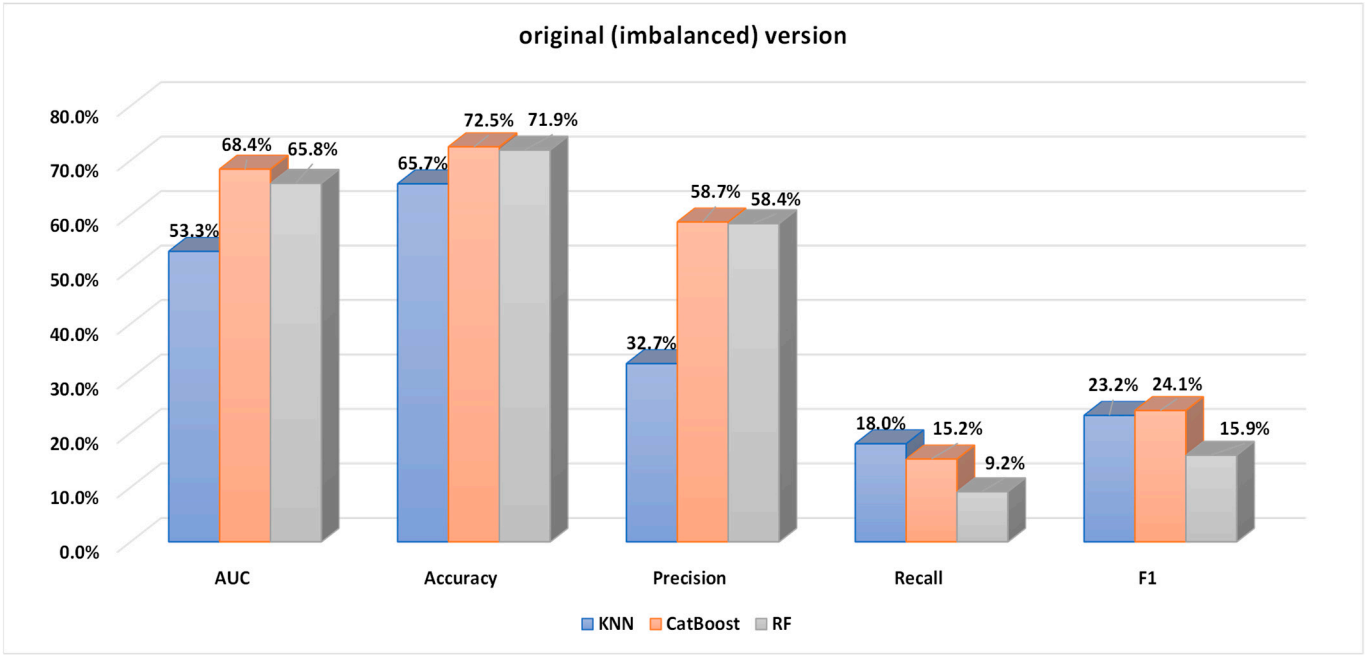
The performance analysis of each individual base classifier of the proposed ensemble model is provided in Section 5.2.1. Besides that, the performance of the proposed model is also studied in Section 5.2.2. Moreover, the performance comparison with the other ensemble learning models is also included.

5.2.1. Performance Analysis of the Base Classifiers

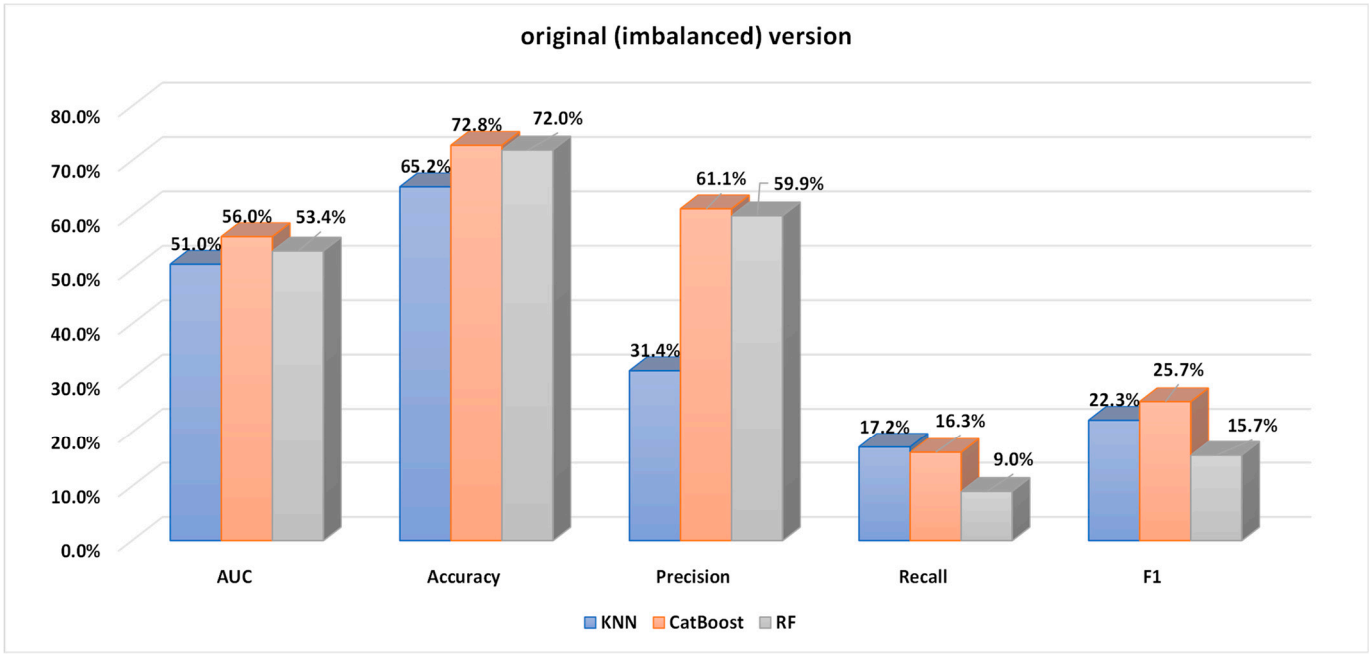
In this section, the performance analysis of each base classifier of the proposed ensemble model is presented. Three scenarios are analyzed for the KNN, CatBoost, and random forest performance comparison. To be specific, the performances of the original (unbalanced data), downsampling (balanced data, performed using the undersampling technique), and oversampling (balanced data, performed using the SMOTE technique) database versions are studied for the three base learners. Figures 6–8 show the evaluation results according to the standard metrics for each learner in the cross-validation and train-test split protocols, respectively, in the three different database versions. On the other hand, Figures 9–11 illustrate the receiver operating characteristic (ROC) plots and precision–recall curves for the base learners in the respective database versions.

In Figures 6–8 (bar plots), we notice a considerable difference between the accuracy score and the F1 score in the original imbalanced-class database version, i.e., the base learners obtain 65% to 73% accuracy scores. In contrast, merely 15% to 26% of the F1 scores are achieved. The accuracy metric is not suitable for imbalanced-class data [35]. We understand that accuracy is the most-used metric to evaluate classification algorithms due to its easy calculation and interpretation. However, when there is a skew in the class distribution (i.e., imbalanced class), the accuracy metric becomes unreliable for evaluating model performance. Specifically, our original database version has ~70% non-churn and ~30% churn. Even when a model fails to predict any churn (this minority class is our interest), its accuracy is still approximately 70% as the data contain ~70% non-churn. Hence, accuracy is misleading if used on imbalanced datasets. In other words, precision, recall, and F1 scores are more appropriate as performance metrics for imbalanced classification problems in the original imbalanced database.

In the figures, we can observe that the sampling/rebalancing techniques are significant in handling imbalanced data classification. The empirical results show that higher precision, recall, and F1 score are obtained in the downsampling and oversampling database versions compared with those in the original imbalanced database version. This indicates that the sampling techniques transform the training dataset to better balance the class distribution to address the negative effects of the imbalanced training dataset, which could be a bottleneck in the performance of various machine learning methods that assume the data distribution to be balanced [36]. In both balanced database versions, the random forest (RF) scores the highest F1 score. RF obtains 67.7% in the cross-validation protocol and 68.2% in the train-test split protocol for the downsampling version, 78% in the cross-validation protocol, and 77.8% in the train-test split protocol for the oversampling version. CatBoost is the second-best-performing machine learning method, followed by the KNN classifier.

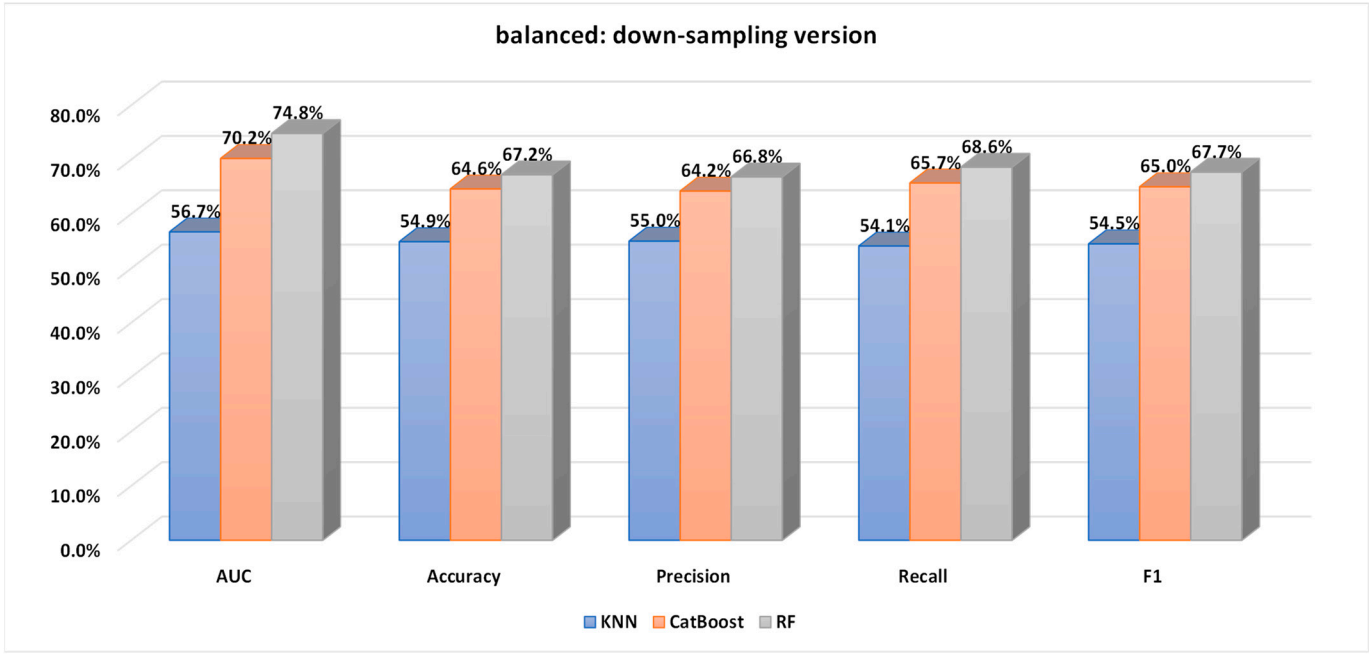


(**a**)

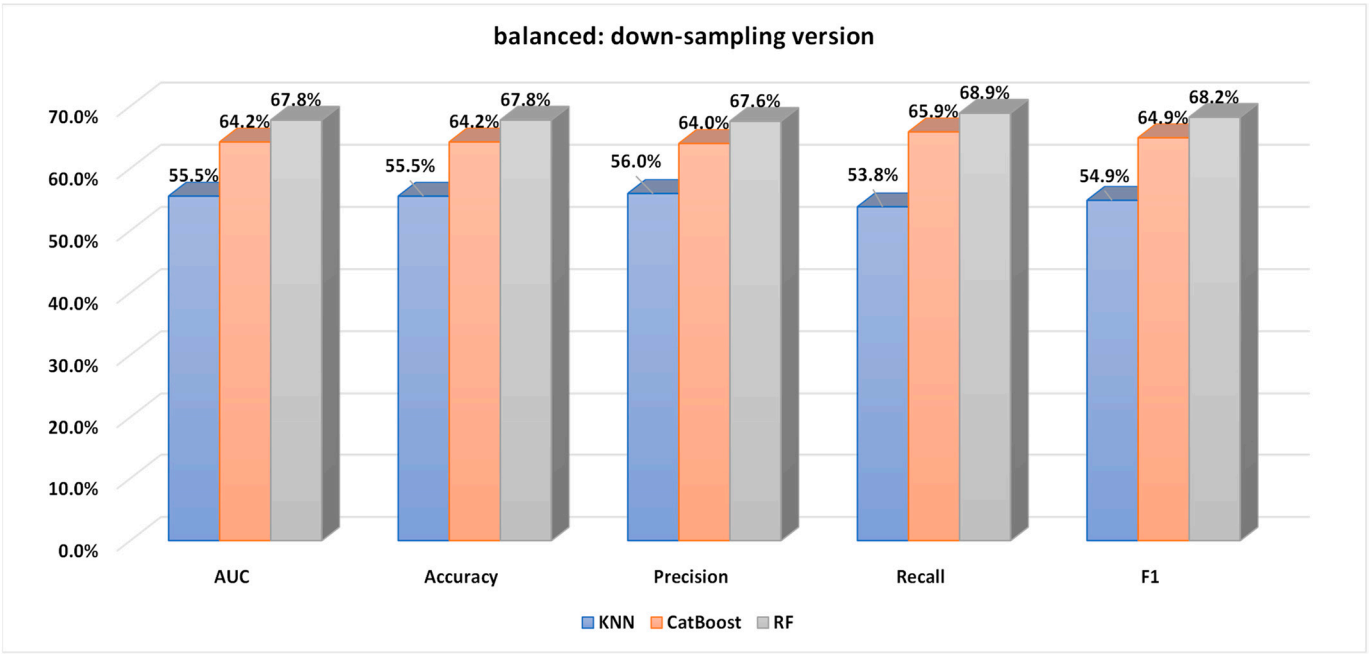


(**b**)

Figure 6. The performance of the base learners in the (**a**) cross-validation protocol and (**b**) train-test split protocol in the original database version.



(**a**)



(**b**)

Figure 7. The performance of the base learners in the (**a**) cross-validation protocol and (**b**) train-test split protocol in the downsampling database version.

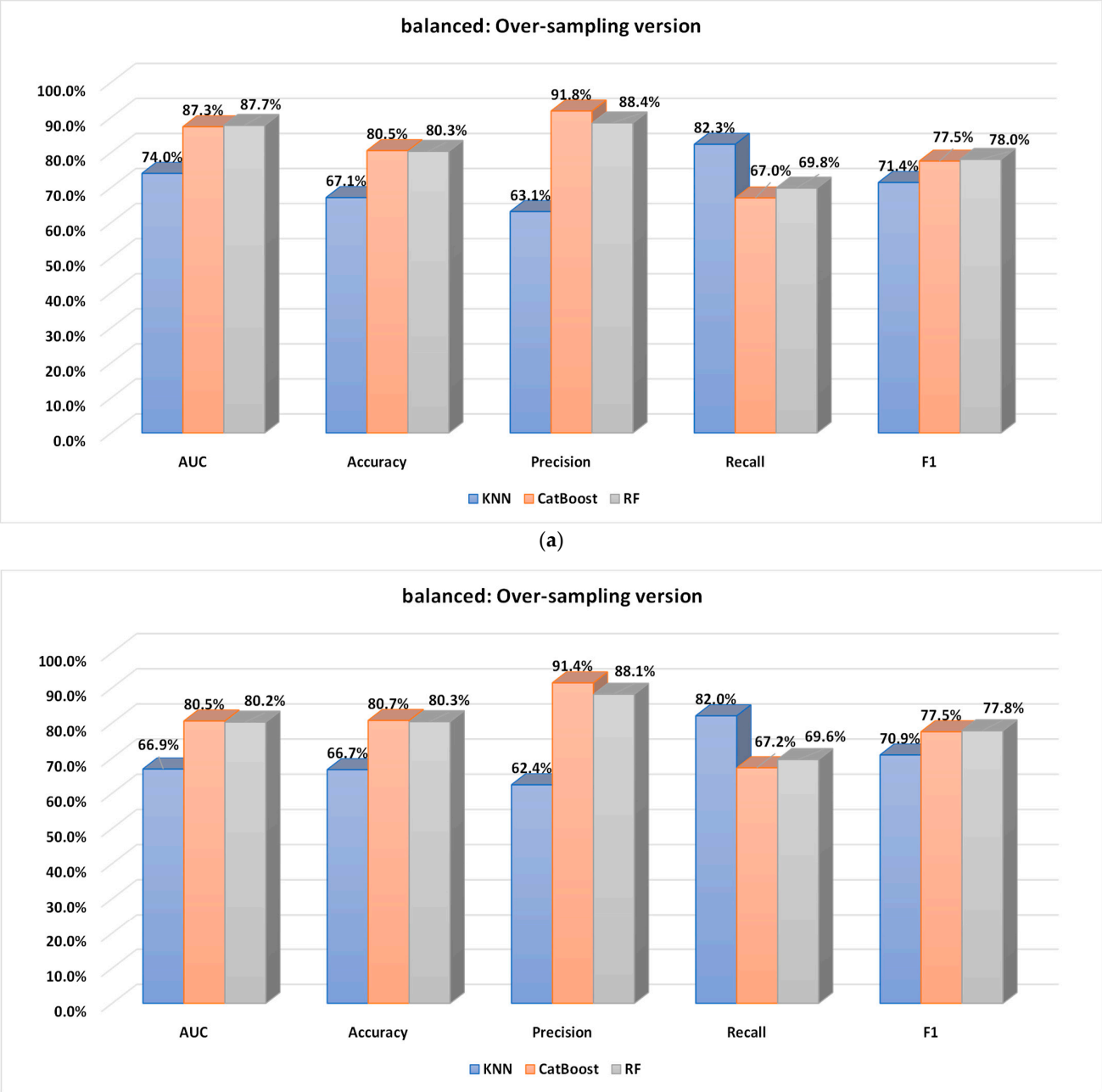


Figure 8. The performance of the base learners in the (**a**) cross-validation protocol and (**b**) train-test split protocol in the oversampling database version.

Figures 9a, 10a and 11a illustrate the ROC curves (plots of true positive rates versus false positive rates) of the three base classifiers. An ROC curve is a misleading visual illustration of imbalanced data and may provide an optimistic view of a model's performance [37]. For example, take a dataset with 10 positives (the minority class) and 10,000 negatives (the majority class). Model X predicts 900 positives, 9 of which are true positives; another model Y predicts 90 positives and 9 of these are true positives. Apparently, model Y has a better performance and is more "precise." However, since ROC analysis measures the true positive rate (TPR) against the false positive rate (FPR), model X obtains $TPR = 9/10 = 0.9$ and $FPR = (900 - 9)/10,000 = 0.0891$; meanwhile, model Y obtains $TPR = 9/10 = 0.9$ and $FPR = (90 - 9)/10,000 = 0.0081$. As expected, identical TPR scores are obtained in both models. However, since the number of negatives mainly dominates that of positives,

the *FPR* difference between both models, i.e., 0.081, is tiny, almost zero. Specifically, a significant change in the number of false positives results in a slight shift in the *FPR*. Therefore, the ROC cannot reflect the actual performance of model Y in the context in which true negatives are not the interest. In our original imbalanced-class database version (see Figure 9a), the AUC is ~0.5, and instead, the average precision score (avg_pcn) from the precision–recall (PR) curve (see Figure 9b) is much lower, i.e., ~0.3, indicating fewer correct predictions. In summary, the PR curve is more appropriate to be used as a metric when the positive class is of more interest than the negative one. This is because precision and recall do not consider true negatives, and a precision–recall curve is not influenced by imbalanced data.

From the precision–recall analysis, we can observe that avg_pcn is higher in the balanced-class database, i.e., the downsampling and oversampling versions, compared with the imbalanced-class version. Between the sampling techniques, the SMOTE over-sampling technique shows superior performance in our case with higher ROC-AUC and PR-avg_pcn scores according to the three classifiers. This is because in the SMOTE technique, the minority class is oversampled by producing synthetic data samples that are marginally different from the original data points. This is accomplished by selecting a data sample from a minority class and locating its *k*-nearest neighbors. Then, new instances are created by interpolating between the selected data sample and its nearest neighbors. These generated synthetic data samples can better represent the data nature. In the subsequent experiments, the SMOTE over-sampling technique was adopted.

Furthermore, we can observe in the precision–recall plots for the downsampling and oversampling versions that the CatBoost and random forest classifiers perform better than the KNN classifiers. The ordered boosting mechanism in the CatBoost classifier contributes to good performance by optimizing the learning objective function. The algorithm constructs a series of decision trees, each of which is trained to correct the errors of the previous tree, enabling the algorithm to capture the data's inherent structure. On the other hand, insights into the significance of each data feature in the random forest classifier help comprehend the underlying nonlinear relationships in the data. This allows the algorithm to identify complex patterns in the data that other methods would have overlooked.

5.2.2. Performance of the Proposed Optimized Weighted Ensemble Learner and Comparison with the Other Ensemble Learning Methods

In this experiment, the proposed optimized weighted ensemble learner's performance is conducted using the train-test split protocol. As aforementioned, the original churn data are preprocessed and sampled to transform the raw data into a clean data format with a balanced class distribution, which can be more effectively processed in machine learning tasks. Three base learners are adopted in this ensemble learner: KNN, CatBoost, and a random forest. The learners' hyperparameters are tuned using grid-search, i.e., GridSearchCV. Weighted soft voting is applied so that each base learner provides a probability score that a specific data sample belongs to a particular target class. The predictions are weighted according to the base learner's importance and summed up. In our proposed ensemble learner, the weights are optimized via Powell's algorithm. Firstly, some initial weight values are chosen, and then the optimization is run. Besides that, the range of each weight is specified with a lower value of zero and an upper value of five so that the search does not operate wildly.

The accuracy, precision, recall, and F1 scores of the proposed optimized weighted ensemble learner are recorded in Table 3. Furthermore, the performances of other classifiers, i.e., a majority voting classifier with the same base learners (i.e., KNN, CatBoost, and a random forest), a uniform weighted soft voting classifier with similar base learners, a stacking classifier with the meta-learner of logistic regression, and an optimized weighted soft voting classifier via the Nelder–Mead algorithm and deep learning models, are also recorded in the table for performance comparison. A stacking classifier is an ensemble learning method that combines multiple base learners/classifiers to create a high-level

meta-learner. The predictions from each base learner are provided as training data to the meta-learner to generate a final prediction.

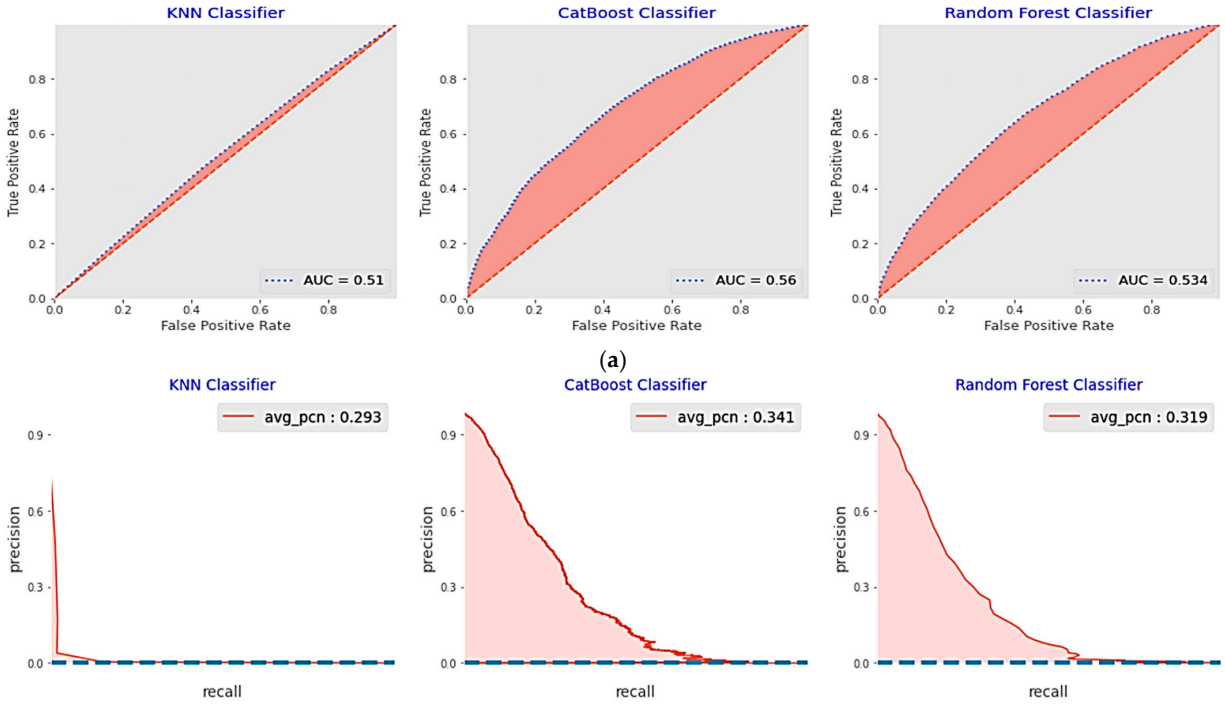


Figure 9. (**a**) ROC plots and (**b**) precision–recall curves for the original database version. (**a**) ROC plots for the original (imbalanced) version; (**b**) precision–recall curves for the original (imbalanced) version.

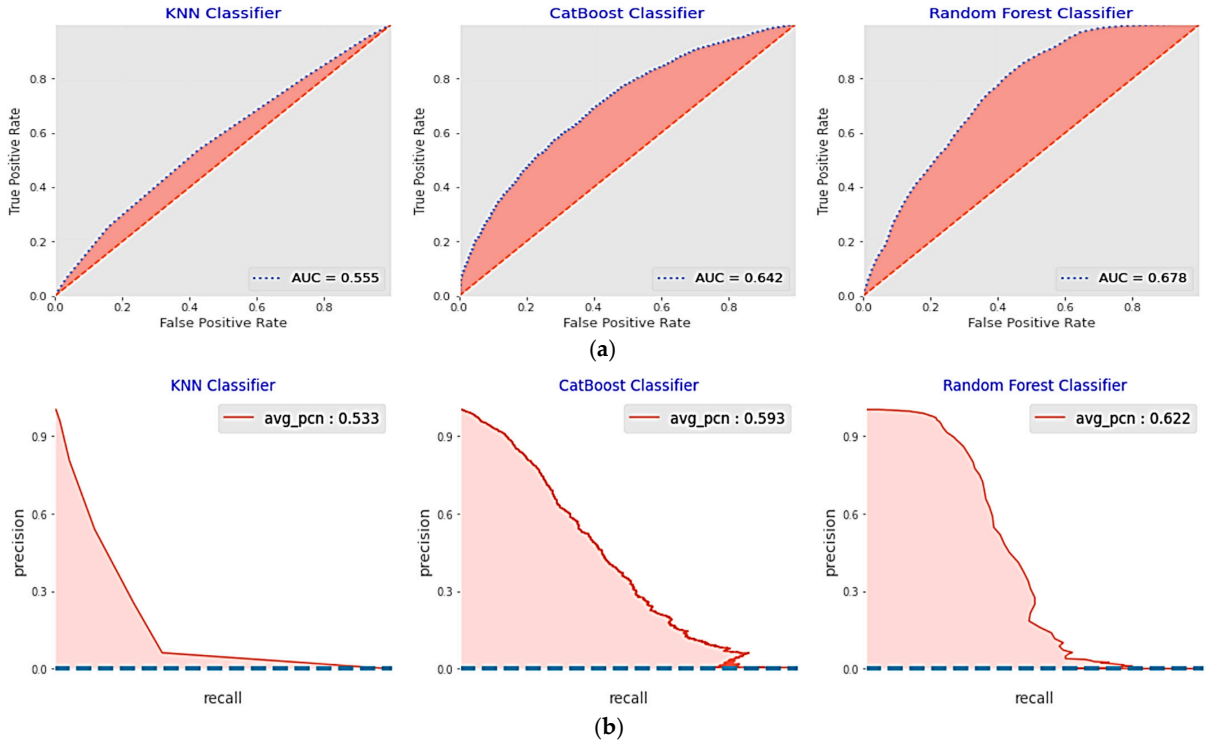


Figure 10. (**a**) ROC plots and (**b**) precision–recall curves for the downsampling database version. (**a**) ROC plots for the downsampling version; (**b**) precision–recall curves for the down-sampling version. Blue dashed line is the precision-recall curve plot of a no-skill classifier.

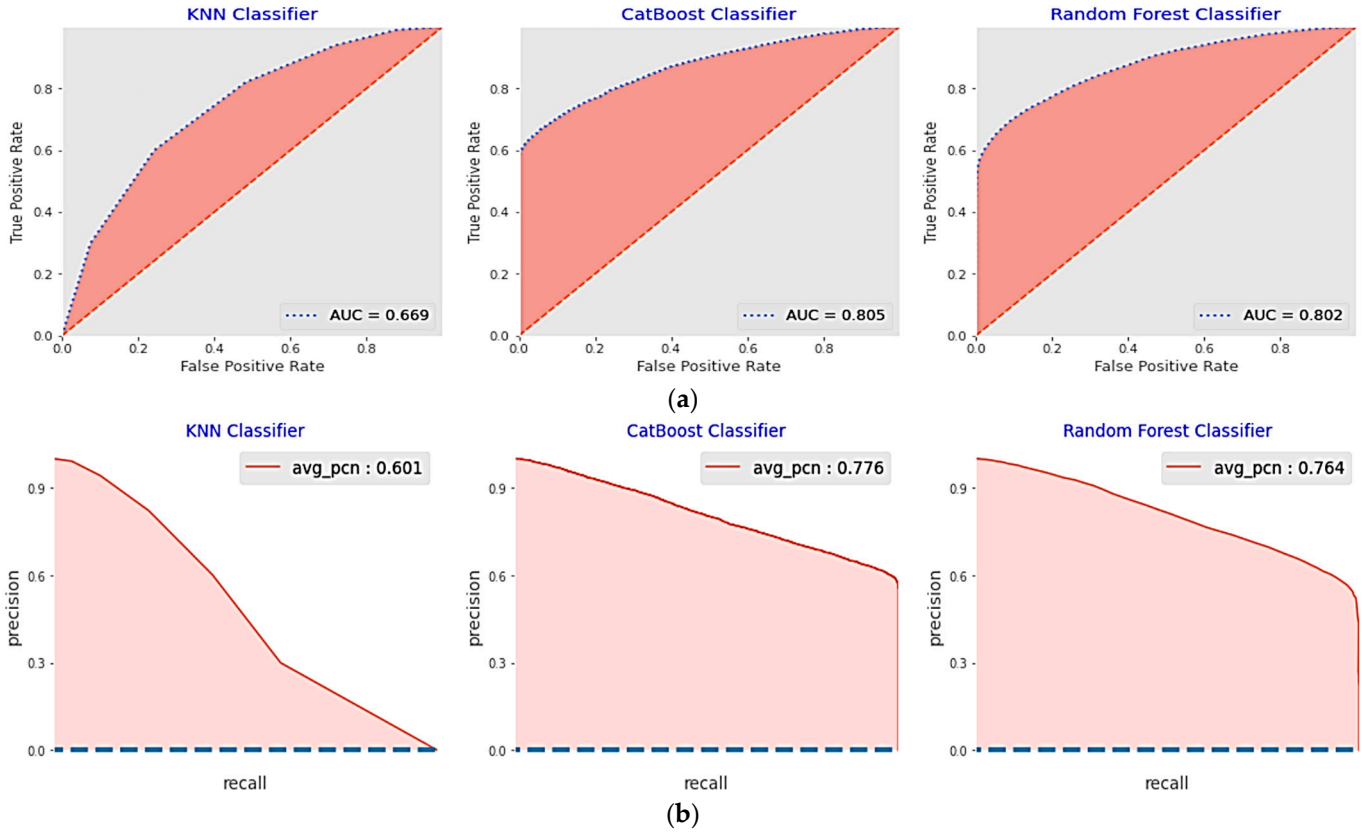


Figure 11. (**a**) ROC plots and (**b**) precision–recall curves for the oversampling database version. (**a**) ROC plots for the oversampling version; (**b**) precision–recall curves for the oversampling version. Blue dashed line is the precision-recall curve plot of a no-skill classifier.

Table 3. Classification performances of the proposed method and the other classifiers.

Classifier	Accuracy	Precision	Recall	F1 Score
Majority/hard voting	0.81975	0.89528	0.71922	0.79765
Soft voting (uniform weight)	0.80826	0.81691	0.78858	0.80250
Stacking classifier with meta learner = logistic regression	0.83674	0.86290	0.79596	0.82808
Weighted soft voting by Nelder–Mead	0.82876	0.91772	0.71769	0.80547
Large feedforward neural network * [22]	0.7166	-	-	-
Convolutional neural network * [22]	0.7166	-	-	-
TL-DeepE * [23]	0.682	-	-	-
Deep-BP-ANN * [24]	0.7938	0.7450	0.8932	0.8124
The proposed weighted ensemble learner with Powell's optimization	0.84114	0.86108	0.80891	0.83418

* Results are obtained from the original papers.

From the empirical results, we can observe that the weighted soft voting classifier with Nelder–Mead's weight optimization has a superior performance to the majority voting and soft voting classifiers due to its ensemble weight optimization. Furthermore, it is also noticed that the stacking classifier performs better than the majority voting and soft voting classifiers as well as the weighted soft voting classifier with Nelder–Mead's weight optimization. This indicates that stacking the individual classifiers helps produce a more reliable prediction. On the other hand, our proposed ensemble learner with Powell's optimization algorithm achieves the highest accuracy with 84.114%. This result signifies the effectiveness of Powell's algorithm for weight optimization, improving the performance of the soft voting ensemble.

When evaluating a churn prediction model, accuracy alone may not be sufficient to assess its effectiveness. Therefore, when determining a churn prediction model, it is crucial

to consider metrics such as precision, recall, and F1 score in addition to accuracy. The table shows that the proposed model scores the highest F1 value at 83% and a high recall value at 81%. Nevertheless, its precision value is lower than the other existing techniques (i.e., the majority voting classifier, stacking classifier, and weighted soft voting classifier with Nelder–Mead's weight optimization). This signifies that the model correctly predicts a higher percentage of customers who are likely to churn. Nevertheless, it also flags some customers who are unlikely to churn as being at risk of churning. In other words, the model may cast a wider net to identify more of the actual churners, which is a crucial objective of a churn prediction model. However, this wider net may result in more false positives, whereby loyal customers are flagged as being at risk of churning, which is reflected in the lower precision. The choice of a churn prediction model depends on the goals and constraints of a business. When the cost of losing a customer to churn is significant, it is critical to identify as many at-risk customers as possible. Hence, a higher recall score is preferred over precision. On the other hand, when the cost of targeting false positives is high (i.e., due to costly retention campaigns), reducing false positives and a lower precision score is preferable.

6. Conclusions

This research on customer churn prediction is significant in helping telecommunication companies to generate huge savings since retaining a churn-probable customer is more cost-efficient than acquiring a new customer. Thus, this research developed a machine learning system to predict the churn of customers in the Cell2Cell telecom company. Efficacious feature engineering and feature transformation were applied to rectify "problematic" data, including those with missing values, noise, and outliers. The data were transformed into an appropriate form for machine learning classification. The presence of a low number of churners creates class imbalance problems. Hence, the SMOTE oversampling technique was implemented to deal with imbalanced data. An optimized weighted ensemble learning model was adopted in this work as the classification model in the proposed system. Specifically, Powell's optimization algorithm was utilized to optimize the model's ensemble weights of influence following the base learners' importance. The empirical results reveal the enhanced ensemble learning system's superiority over the other ensemble learning models. Furthermore, the proposed method exhibits higher accuracy and F1 scores than the deep learning models.

Author Contributions: Conceptualization, W.H.K. and Y.H.P.; methodology, Y.H.P.; software, Y.H.P.; validation, S.Y.O., L.-Y.-K.W. and Q.W.P.; formal analysis, Y.H.P.; investigation, W.H.K.; resources, W.H.K.; data curation, W.H.K.; writing—original draft preparation, Y.H.P. and W.H.K.; writing—review and editing, Y.H.P. and S.Y.O.; visualization, L.-Y.-K.W.; supervision, Y.H.P.; project administration, Y.H.P. and Q.W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Matuszelański, K.; Kopczewska, K. Customer Churn in Retail E-Commerce Business: Spatial and Machine Learning Approach. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 9. [CrossRef]
2. Ullah, I.; Raza, B.; Malik, A.K.; Imran, M.; Islam, S.U.; Kim, S.W. A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. *IEEE Access* **2019**, *7*, 60134–60149. [CrossRef]
3. Verbeke, W.; Martens, D.; Mues, C.; Baesens, B. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Syst. Appl.* **2011**, *38*, 2354–2364. [CrossRef]

4. Jessica Tracking Churn To Measure Customer Loyalty and Satisfaction in the Telecommunications Industry | Open World Learning. Available online: https://www.openworldlearning.org/tracking-churn-to-measure-customer-loyalty-and-satisfaction-in-the-telecommunications-industry/ (accessed on 2 February 2023).
5. Sharma, A.; Shukla, P.; Gourisaria, M.K.; Sharma, B.; Dhaou, I.B. Telecom Churn Analysis using Machine Learning in Smart Cities. In Proceedings of the 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 23–25 January 2023; pp. 1–5. [CrossRef]
6. Sharma, A.; Gupta, D.; Nayak, N.; Singh, D.; Verma, A. Prediction of Customer Retention Rate Employing Machine Learning Techniques. In Proceedings of the 2022 1st International Conference on Informatics, Noida, India, 14–16 April 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022; pp. 103–107.
7. Huang, B.; Kechadi, M.T.; Buckley, B. Customer churn prediction in telecommunications. *Expert Syst. Appl.* **2012**, *39*, 1414–1425. [CrossRef]
8. Kimura, T. Customer churn prediction with hybrid resampling and ensemble learning. *J. Manag. Inf. Decis. Sci.* **2022**, *25*, 1–23.
9. Kim, S.; Lee, H. Customer Churn Prediction in Influencer Commerce: An Application of Decision Trees. *Procedia Comput. Sci.* **2022**, *199*, 1332–1339. [CrossRef]
10. ÜNLÜ, K.D. Predicting credit card customer churn using support vector machine based on Bayesian optimization. *Commun. Fac. Sci. Univ. Ankara Ser. A1 Math. Stat.* **2021**, *70*, 827–836. [CrossRef]
11. Thorat, A.S.; Sonawane, V.R. Customer Churn Prediction in the Telecom Industry Using Machine Learning Algorithms. *Comput. Integr. Manuf. Syst.* **2023**, *29*, 1–11.
12. Bose, A.; Thomas, K.T. A Comparative Study of Machine Learning Techniques for Credit Card Customer Churn Prediction. In *Lecture Notes on Data Engineering and Communications Technologies*; Springer Science and Business Media Deutschland GmbH: Berlin/Heidelberg, Germany, 2023; Volume 141, pp. 295–307.
13. Coussement, K.; De Bock, K.W. Customer churn prediction in the online gambling industry: The beneficial effect of ensemble learning. *J. Bus. Res.* **2013**, *66*, 1629–1636. [CrossRef]
14. Jain, H.; Khunteta, A.; Srivastava, S. Churn Prediction in Telecommunication using Logistic Regression and Logit Boost. *Procedia Comput. Sci.* **2020**, *167*, 101–112. [CrossRef]
15. Lalwani, P.; Mishra, M.K.; Chadha, J.S.; Sethi, P. Customer churn prediction system: A machine learning approach. *Computing* **2022**, *104*, 271–294. [CrossRef]
16. Elgohary, E.M.; Galal, M.; Mosa, A.; Elshabrawy, G.A. Smart evaluation for deep learning model: Churn prediction as a product case study. *Bull. Electr. Eng. Inform.* **2023**, *12*, 1219–1225. [CrossRef]
17. Xu, S.; Tang, Q.; Jin, L.; Pan, Z. A cascade ensemble learning model for human activity recognition with smartphones. *Sensors* **2019**, *19*, 2307. [CrossRef]
18. Tariq, M.U.; Babar, M.; Poulin, M.; Khattak, A.S. Distributed model for customer churn prediction using convolutional neural network. *J. Model. Manag.* **2022**, *17*, 853–863. [CrossRef]
19. Gabhane, M.D.; Suriya, A.; Kishor, S.B. Churn Prediction in Telecommunication Business using CNN and ANN. *J. Posit. Sch. Psychol.* **2022**, *2022*, 4672–4680.
20. Sudharsan, R.; Ganesh, E.N. A Swish RNN based customer churn prediction for the telecom industry with a novel feature selection strategy. *Connect. Sci.* **2022**, *34*, 1855–1876. [CrossRef]
21. Mishra, A.; Reddy, U.S. A Novel Approach for Churn Prediction Using Deep Learning. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017. [CrossRef]
22. Umayaparvathi, V.; Iyakutti, K. Automated Feature Selection and Churn Prediction using Deep Learning Models. *Int. Res. J. Eng. Technol. (IRJET)* **2017**, *4*, 1846–1854.
23. Ahmed, U.; Khan, A.; Khan, S.H.; Basit, A.; Haq, I.U.; Lee, Y.S. Transfer Learning and Meta Classification Based Deep Churn Prediction System for Telecom Industry. *arXiv* **2019**, arXiv:1901.06091. [CrossRef]
24. Wael Fujo, S.; Subramanian, S.; Ahmad Khder, M.; Fujo, W.; Khder, A. Customer Churn Prediction in Telecommunication Industry Using Deep Learning. *Inf. Sci. Lett.* **2022**, *11*, 185–198. [CrossRef]
25. Dorogush, A.V.; Ershov, V.; Yandex, A.G. CatBoost: Gradient Boosting with Categorical Features Support. *arXiv* **2018**, arXiv:1810.11363.
26. Zhu, M.; Liu, J. Telecom Customer Churn Prediction Based on Classification Algorithm. In Proceedings of the 2021 International Conference on Aviation Safety and Information Technology, Changsha China, 18–20 December 2021; ACM: New York, NY, USA, 2021; pp. 268–273.
27. Sagala, N.T.M.; Permai, S.D. Enhanced Churn Prediction Model with Boosted Trees Algorithms in the Banking Sector. In Proceedings of the 2021 International Conference on Data Science and Its Applications, ICoDSA 2021, Bandung, Indonesia, 6–7 October 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 240–245.
28. Ibrahim, A.A.; Ridwan, R.L.; Muhammed, M.M.; Abdulaziz, R.O.; Saheed, G.A. Comparison of the CatBoost Classifier with other Machine Learning Methods. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 738–748. [CrossRef]
29. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Statistics.* **2001**, *29*, 1189–1232. [CrossRef]

30. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018): Advances in neural Information Processing Systems, Montréal, QC, Canada, 2–6 December 2018; pp. 6638–6648.
31. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2012.
32. Taha, A. Intelligent Ensemble Learning Approach for Phishing Website Detection Based on Weighted Soft Voting. *Mathematics* **2021**, *9*, 2799. [CrossRef]
33. Mathews, J.H. *Module for Powell Search Method for a Minimum*; Fullerton Retrieved 16 June 2017; California State University: Northridge, CA, USA, 2017.
34. Zhang, S.; Zhou, Y. Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis. *Discret. Dyn. Nat. Soc.* **2015**, *2015*, 4813360. [CrossRef]
35. Gu, Q.; Zhu, L.; Cai, Z. Evaluation measures of the classification performance of imbalanced data sets. In *ISICA 2009: Computational Intelligence and Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 51, pp. 461–471.
36. Goel, G.; Maguire, L.; Li, Y.; McLoone, S. Evaluation of sampling methods for learning from imbalanced data. In *ICIC 2013: Intelligent Computing Theories*; Lecture Notes in Computer Science Book Series; Springer: Berlin/Heidelberg, Germany, 2013; pp. 392–401. [CrossRef]
37. Branco, P.; Torgo, L.; Ribeiro, R.P. A Survey of Predictive Modelling under Imbalanced Distributions. *arXiv* **2015**, arXiv:1505.01658.