processes

MDPI

Article

Solving the Problem of Class Imbalance in the Prediction of Hotel Cancelations: A Hybridized Machine Learning Approach

Mohd Adil [1], Mohd Faizan Ansari [2], Ahmad Alahmadi [3], Jei-Zheng Wu [4,*] and Ripon K. Chakrabortty [5]

[1] Department of Management Studies, NIT Hamirpur, Hamirpur 177005, India; adil.dms@nith.ac.in
[2] Department of Computer Science, Aligarh Muslim University, Aligarh 202002, India; mfansari2395@gmail.com
[3] Department of Electrical Engineering, College of Engineering, Taif University, Taif 21944, Saudi Arabia; aziz@tu.edu.sa
[4] Department of Business Administration, Soochow University, Taipei 100, Taiwan
[5] Capability Systems Centre, School of Engineering and IT, UNSW Canberra at ADFA, Canberra, ACT 2612, Australia; r.chakrabortty@adfa.edu.au
* Correspondence: jzwu@scu.edu.tw

Abstract: The cancelation of bookings puts a considerable strain on management decisions in the case of the hospitability industry. Booking cancelations restrict precise predictions and are thus a critical tool for revenue management performance. However, in recent times, thanks to the availability of considerable computing power through machine learning (ML) approaches, it has become possible to create more accurate models to predict the cancelation of bookings compared to more traditional methods. Previous studies have used several ML approaches, such as support vector machine (SVM), neural network (NN), and decision tree (DT) models for predicting hotel cancelations. However, they are yet to address the class imbalance problem that exists in the prediction of hotel cancelations. In this study, we have shortened this gap by introducing an oversampling technique to address class imbalance problems, in conjunction with machine learning algorithms to better predict hotel booking cancelations. A combination of the synthetic minority oversampling technique and the edited nearest neighbors (SMOTE-ENN) algorithm is proposed to address the problem of class imbalance. Class imbalance is a general problem that occurs when classifying which class has more examples compared to others. Our research has shown that, after addressing the class imbalance problem, the performance of a machine learning classifier improves significantly.

Keywords: machine learning; class imbalance; hotel cancelation; SMOTE-ENN

Citation: Adil, M.; Ansari, M.F.; Alahmadi, A.; Wu, J.-Z.; Chakrabortty, R.K. Solving the Problem of Class Imbalance in the Prediction of Hotel Cancelations: A Hybridized Machine Learning Approach. *Processes* **2021**, *9*, 1713. https://doi.org/10.3390/pr9101713

Academic Editor: Chien-Chih Wang

Received: 16 August 2021
Accepted: 21 September 2021
Published: 24 September 2021

1. Introduction

Revenue administration is the application of data frameworks and estimating schemes, and it is employed to assign correct proportions to an appropriate client at a genuine price [1]. It was initially created in 1966 by the aircraft industry [2] and was subsequently embraced by more service provider businesses, such as hotels, rental cars, golf courses, and casinos [1,2]. In the hospitality industry (rooms division), the definition of revenue administration is "making the right room available for the right person and the genuine price at the apparent time via the right circulation medium" [3]. Considering that lodgings (hotels) have an established number of rooms, and that they offer them as a perishable item to provide the right room to a suitable individual, lodgings have to acknowledge appointments ahead of time. Booking is a kind of an agreement between a lodging and its clients [4], and it gives clients the right to cancel an agreement. For hotels, bookings in advance are the main indicator of a hotel's forecast performance [5]. However, cancelations impact hotels more than guests, as a hotel should have rooms for clients who respect their bookings but, at the same time, it struggles financially when a client cancels a booking or does not show up [4]. A booking cancelation occurs when a client closes their contract

before their entry, while a no-show is when a client does not inform the lodging of a change in plans and fails to check in.

However, booking may also be canceled due to some apprehensible reason such as bad weather, vacation rescheduling, sudden illness, change in meeting place, and many others. However, [6,7] pinpointed that, currently, a sizeable number of cancelations occur because of deal-seeking clients who seek out the best bargain. Occasionally, these customers keep looking for a better deal for the same service or product, even after they have booked. In a few cases, clients indeed make additional bookings to secure their alternatives and, after that, cancel all but one [4]. As a result, cancelations have a compelling effect on demand administration choices within a revenue administration framework.

While exact predictions are a rigid instrument in terms of revenue administration performance, predictions are, without a doubt, influenced by cancelations [4]. Booking cancelations can comprise up to 20% of all bookings acknowledged by a given lodging [8], and this can rise to 60% in the case of airplane terminal and roadside lodgings [9]. However, with such a large cancelation rate, to mitigate loses, hotel managers have implemented many overbooking strategies and restrictive cancelation policies [3,4,10]. However, such strategies can have a negative effect on a hotel's revenue, as well as its social image. For illustration, overbooking can spur a lodging to deny renumeration to a client, which can influence the latter's perception of the lodging and persuade them to seek another lodging [11]. Restrictive cancelation policies, particularly non-refundable and 48 h advance cancelation deadlines [10], decrease client bookings, as well as income, due to the application of impressive cost rebates and the number of bookings [6,10].

In machine learning (ML), supervised learning is ordinarily partitioned into two sorts of problem [11]: "regression", when an output is quantitative (e.g., stock market prediction), or "classification", when an output is categorical or discrete (e.g., forecasting in the case of hotel bookings that show whether a customer "will cancel booking" or "will not cancel booking"). Evidently, several studies in the existing literature have already proposed strategies to relieve the consequence of cancelations in terms of revenue and stock allotment, cancelation arrangements, and overbookings [5,12,13]. However, most of the published research focused on the carrier industry, which differentiates itself from the hospitality industry from a number of perspectives [14–19]. For instance, in the carrier industry, the demand forecast is used to determine the number of seats under a particular class (like economy, business, and semi-business class) [16]. Furthermore, in the carrier industry, the task is to predict the optimal limits on the number of bookings that can book for a particular reservation class, whereas in hotel booking, tourists book for a separate room according to their budget and the facilities they are looking for [14]. Given this, in the hospitality industry, external factors such as—location, weather condition, visiting place etc.—plays an important role; however, in the carrier industry, these factors do not have much importance. However, in recent years, research related to the hospitality industry has gained wider attention [20,21]. Most research has used traditional statistical methods such as regression [9], whereas some research has used the advantages conferred by machine learning methods and techniques [21]. A similar plan applies to the exploration of demand forecasting to anticipate retractions, particularly in relation to hospitality [8,9,22,23]. Moreover, only three investigations have utilized information specific to lodgings (property management systems—PMS information) [9,22,24]. Furthermore, the other two investigations utilized passenger name record (PNR) information, which is an aircraft industry standard set up by the International Air Transport Association (International Civil Aviation Organization, 2010).

Much of the literature has also assumed booking cancelation to be a "regression problem". However, the prediction of hotel cancelations using machine learning is limited, and only a few studies have considered it a classification problem [22,24,25]. In fact, authors in [8] specified that "it is hard to say that one can predict whether a booking will be canceled or not with high accuracy". Moreover, António et al. [24] presented that it is possible to predict hotel cancelations as a classification problem using machine learning

approaches, and they achieved high accuracy in their study. They evaluated a set of machine learning classifiers for four separate resort hotels in Portugal. Authors in [25] checked the effectiveness of machine learning models in a real environment, and they built a prototype model with computerized AI and intended to search property management systems (PMS) information from past forecast hits and mistakes.

Since booking cancelations can be solved as regression and classification problems, it is important to know when to choose between these two methods. For instance, when the only aim is to estimate cancelation rates, then it can be considered a regression problem; however, when the aim is to estimate the likelihood of a booking being canceled and to understand the reason for such a cancelation, it should be considered a classification method [26]. Furthermore, classification allows for the estimation of an overall cancelation rate [26]. Another reason to consider booking cancelation as a classification problem is that, from class output, it is also possible to achieve a quantitative output [24]. For instance, in [24], the authors suggested that the number of bookings predicted as "will cancel" on a certain day can be removed from the demand to achieve the net demand, while cancelation rates can be calculated by dividing the total bookings predicted as "will cancel" by the total number of bookings for a certain day. In this study, we also consider hotel cancelation as a classification problem.

Moreover, in ML, classification algorithms consider that every class has an equal number of examples, which, in practice, may also fail due to class imbalances. In an imbalanced dataset, the class with fewer examples is called a minority class, and the category with many examples is called a majority class. Machine learning algorithms that use imbalanced datasets overlook this imbalanced distribution of classes that ultimately results in poor performance for the minority class (because a model will learn more about the majority class during classifier training, creating model bias for the majority class) [27]. In terms of hotel booking cancelations, the minority class is classified by its "will cancel booking" attitude; thus, if we train classifiers on imbalance data for hotel booking cancelations, the classifiers will mostly learn about the majority class, or the "will not cancel booking" class. This erroneous information can have a significant effect on a hotel's revenue and reputation, as, in most cases, hotel administrators assume that a particular booking will not cancel, since the classifier is trained in a certain way to demonstrate that a particular booking will not be canceled; in reality, however, the opposite might occur. As a classifier trained on an imbalanced dataset can become a challenge for hotel administrators, and they are therefore unable to properly track which booking might cancel; actions are required to generate revenue for the hotel and manage the image of said hotel in the eyes of their customers. This imbalanced distribution of classes also exists in hotel booking cancelation classifications. This question has not been addressed in previous studies, and there is a need to address it so that hotel administrators can create better policies and take certain actions to increase revenue.

To overcome the abovementioned shortcomings, this study introduces a synthetic minority oversampling technique and an edited nearest neighbors (SMOTE-ENN) algorithm to address the issue of class imbalance in the case of hotel booking cancelations. This algorithm first generates the examples for a minority class with the help of SMOTE. Thereafter, it uses the neighborhood noise removing rule based on the edited neighbor (ENN) [28] to discard the extra overlaying between classes, which eliminates samples that vary from two examples in the three closest neighbors [29]. Therefore, the methodological contribution of this research is the introduction of SMOTE-ENN to address the problem of class imbalance in the case of hotel booking cancelations, i.e., the associations between over-sampling and under-sampling techniques. By over-sampling, it creates examples for the minority class and discards the noise from the dataset using the ENN under-sampling technique. In this research, we present a hybrid approach that combines the oversampling method and a machine learning algorithm for hotel cancelation predictions. Our approach first utilizes the SMOTE-ENN to adjust class distributions. Next, it uses machine learning algorithms for hotel cancelation predictions. The first experiment was conducted to normal-

ize the data. The second experiment balanced the class distribution using SMOTE-ENN. A comparison between proposed and current methodologies is assessed in the third experiment. Furthermore, we also used feature selection and feature engineering for selecting important features that have greater impact in prediction for further improvements. The remainder of this composition is characterized as follows: Section 2 presents a literature review related to the hospitality industry and hotel cancelations. Section 3 presents a procedure for hotel cancelation predictions, which initially sums up the trial dataset and our oversampling method (SMOTE-ENN). As the fundamental contribution of this study, Section 3 presents the hybrid approach for hotel cancelation predictions. In Section 4, we show the experimental results of the study and compare them with existing methods. Section 5 presents the conclusion of the study. Finally, implications, limitations and future research issues are presented in Sections 6 and 7, respectively.

2. Related Works

Booking cancelation is a well-known issue in revenue administration, and it is applicable to the service industry and, most importantly, to the hospitality industry. Customers' increasing interest in the internet has changed the way in which they buy or look for any service. Current customer behavior has a considerable influence on contemporary research on the issue of booking cancelations, particularly that related to the effects of cancelations on revenue and inventory allocation, as well as on cancelation and overbooking policies [12,13]. That said, there is minimal literature related to booking cancelations in the hospitality industry. For instance, authors in [23] presented a neural network model and a regression neural network model for predicting customer cancelations. Their study showed that both prediction models achieved good prediction capabilities and could be useful in service capacity scheduling. Authors in [20] used competitive sets, a recursive approach for forecasting daily occupancy in a hotel. Other authors in [30] applied a linear approximation technique to decide price and seat control simultaneously in the airline industry. Authors used a data mining method to forecast cancelations at any time, and they addressed the behavior of customers in different stages of booking [8].

With rapid advancements in affordable data storage, huge amounts of data availability, less expensive, and more powerful computing have all contributed to the success of ML [26]. In turn, this has motivated industries to develop robust ML models for analyzing big and complex data simultaneously [27]. Machine learning tools facilitate the identification of beneficial liberties and risks [28], making ML use progress rapidly and strengthening the employment of ML in nearly every field [29]. However, in the case of hotel cancelations, there are only a limited number of studies that have utilized ML algorithms. For instance, authors utilize data science methods to synthesize the current fining of booking cancelations in travel- and tourism-related industries, and they have identified a new topic related to booking cancelation research [31]. Authors have also employed big data to improve hotel demand and its deviation from booking cancelations [32]. Their study suggests that, by identifying cancelation factors, this model helps hotel management understand cancelation patterns and allows them to make changes or adjustments in a hotel's cancelation policies and tackle overbooking according to clients' booking behaviors. Other authors have addressed hotel cancelation as a classification problem, and their study shows that a classification model can achieve suitable accuracy [24]. They included four hotels in their study to predict hotel cancelation rates. They presented an automated machine learning-based support system to predict hotel booking cancelations, developing two prototypes and observing their performance. Their system was able to allow hotels to predict overall demand, which helps hotels to make better decisions and act on which booking should be accepted or rejected, as well as to make key changes in booking and room prices.

None of the previous studies explored the issue of imbalance in hotel cancelation predictions. As such, in this research, we combined the imbalanced SMOTE-ENN method with a machine learning classifier to predict hotel booking cancelation patterns.

3. Methods

In this study, we introduce an oversampling (SMOTE-ENN) method to address the class imbalance issue in hotel cancelation predictions. We used the random forest (RF) classifier to train and predict hotel cancelation. Our proposed approach has significantly increased the performance of the RF classifier. In this section, we formulate the proposed methodology for predicting hotel cancelation. Figure 1 represents the overall structure of the proposed methodology. In the first step, it takes the dataset and performs some of the necessary data pre-processing; in the next step, feature selection and feature engineering are performed. Feature selection is performed to select the imported features that have more influence on prediction, while feature engineering is performed to create other features from existing features, which can have a positive impact on classifier performance. After feature selection and engineering, the dataset goes to the random forest machine learning classifier, where it learns the relationship between different features and predicts whether a client will cancel their hotel reservation. We trained a random forest classifier on the train set and accessed its performance on the test set. RF is a classification algorithm with a set of several decision trees. A detailed description of a decision tree and its working can be found in [33]. Each tree in the forest gives a class score, and the class that achieves the most votes becomes the final prediction. The random forest algorithm works in the following manner: First, it selects random samples from the dataset; next, it creates decision trees for every sample and provides the prediction; then, it performs a voting step for each prediction; in the last step, it selects the prediction that received the most votes.

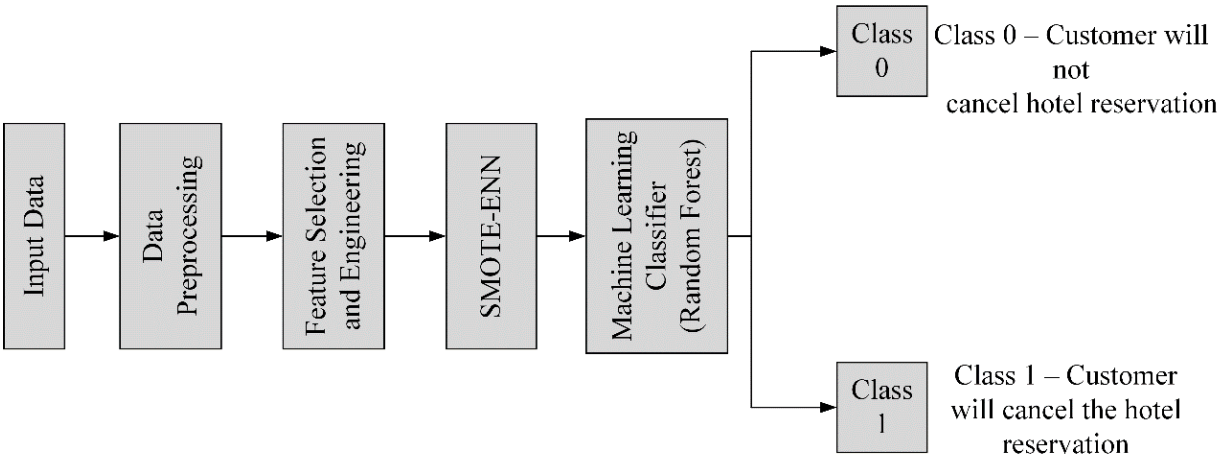

Figure 1. Representation of the conceptual methodology.

3.1. Dataset Description and Understanding

Datasets for this study were collected from [24]; the authors collected data from a Portuguese hotel chain that agreed to provide access to the PMS data for their two hotels. One of their hotels was a resort hotel (H1), while another was a city hotel (H2). Both are considered four-star hotels with an availability of over 200 rooms. They collected data from July 2015 to August 2017; however, for the H2 hotel, the authors used data from September, since this hotel was engaged in a soft opening process. We have also included the same data in our study. Figure 2 shows the cancelation percentage for the resort hotel and the city hotel; we can observe from the figure that the city hotel had a greater number of cancelations compared to the resort hotel.

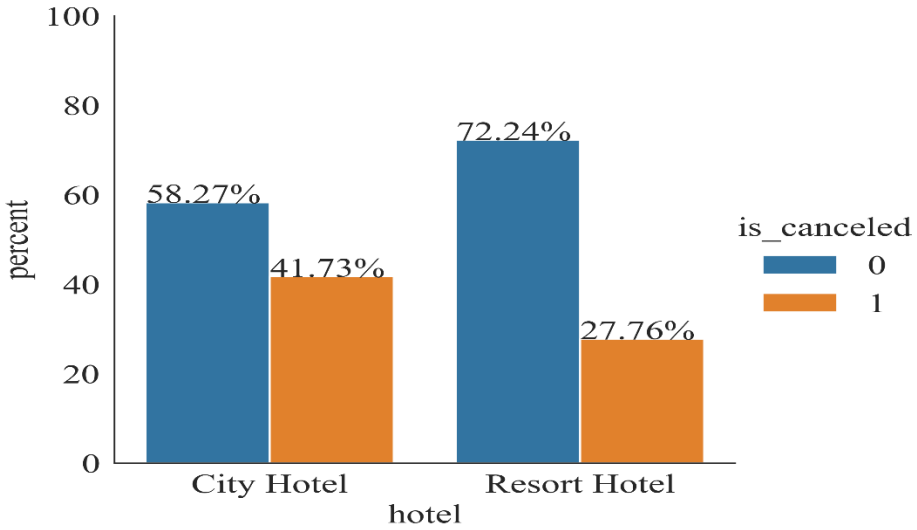

Figure 2. Distribution of examples for hotels H1 and H2.

3.2. Feature Selection and Engineering

Feature selection and feature engineering are essential steps in an ML problem [34–36]; they not only require technical knowledge, but also need domain knowledge and intuition [37,38]. The success of any ML project relies on feature selection and feature engineering. We removed some features that were not imported and created some new features from the existing features that significantly improved the performance of the classifier. This transformation in the dataset showed the importance of feature selection and engineering. First, we removed the company, agent, and country columns from the dataset, since the company column was missing more than 90% of its values. Next, we removed the agent column, as 13% of its values were missing; there were 333 unique agents (too many agents), which may not be predictable. Additionally, NaN values could be the agents that were not listed among the 333 unique agents. We could not predict agents and, since we were missing 13% of the agents' values among all data, we decided to discount this column. We also removed the country column since it introduced spillage in the model [24]; spillage was due to the fact that Portugal was a default nation of root that was confirmed and corrected at check-in [24].

We modified some of the existing features present in the dataset. For example, we created stay_night as a sum of Stays_in_week _night and stays_in_weeked_night. We created a bill feature, which is the multiplication of stays night and adr; this feature contributed significantly to classifier performance, as we looked after generating a correlation matrix. We renamed assigned_room_type and reserved_room_type as room_assignment, since each column represented the same thing, and we removed these columns before feeding our data into the classifier. We converted deposite_type object column into numerical column by fill no_deposit and refundable column with 0 and non_refund column with 1. We created an is_family column by applying a logical operation on the adults, children, and babies column. In addition to this, we made a new column, total_customer, by combining the adult, children, and babies columns and removing it from the final dataset. We also removed reservation_status_date, arrival_date_week_number, arrival_date_month, arrival_date_year, and arrival_date_day_of_month because they were less important in terms of predictions. Finally, we removed the reservation_status column, since it was highly correlated with the predicting column. Table 1 shows the list of original features and derived features after the data selection and data engineering column.

Table 1. Description of feature column after feature selection and engineering.

Feature Name	Original/Modified	Description
Is_canceled	Categorical	Outcome feature: showing whether the booking was canceled (0: no; 1: yes)
Lead_time	numeric	Number of days before appearance that the booking was set in the hotel
Stays_in_weekend_nights	Numeric	From the entire evening, how many were in ends of the week (Saturday and Sunday)
Stays_in_week_nights	Numeric	From the entire evening, how many were during workdays (Monday to Friday)
Is_repeated_guest	categorical	Binary value indicating whether a customer was a repeated guest at the time of booking (0: No, 1: Yes), created by comparing the time of booking with the guest history creation record
Previous_cancelations	Numeric	Total of previous bookings that were canceled by the client
Previous_bookings_not_canceed		Total of previous bookings that were not canceled by the client
Booking_changes	Numeric	Heuristic made by adding the count of booking changes (corrections) earlier to the entry that may show cancelation behavior (arrival or departure dates, number of people, type of meal, ADR, or reserved room type)
Days_in_waiting_list	Numeric	Count of booked days was shown in list before it was affirmed
Adr	Numeric	Average daily rate
Required_car_parking_space	Numeric	Total car parking spaces a visitor required
Total_of_special_requests	Numeric	Total extraordinary demands made by a client
Stay_nights (Derived)	Numeric	Total number of nights stays in hotel
Bill (Derived)	Numeric	Multiplication of stays night and average daily rate
Is_family (Derived)	Categorial	Based on the logical operation whether visiting customer was whole family, a couple, or single
Total_customer (Derived)	Numeric	Sum of the adults, children, and babies
Deposit_given (Modified)	Categorical	Modified from deposit type column
Meal	Categorical	ID of meal guest
Market_segment	categorical	Group of segments to which the booking was assigned
Distribution channel	Categorical	Name of the medium used to make booking
Room assignment (Modified)	Categorical	Room type assigned to a customer
Customer type		sort of client (group, contract, transitory, or temporary party who required more than one room)

3.3. SMOTE-ENN

After feature selection and feature engineering, we applied an oversampling and under-sampling algorithm (SMOTE-ENN) to address the issue of class imbalance. This method uses SMOTE oversampling of the minority class and edited nearest neighbors (ENN) under-sampling (or cleaning) of the majority class to produce a better proportion of each class so that the model learns better and does not have bias towards the minority class. The proposed SMOTE-ENN method also addresses overfitting issues, which happen due to the stand-alone SMOTE, which creates too many exact copies of the minority class (or oversampling). If there are a small number of examples for the minority class, the classifier suffers from overfitting problems [39]. This method first uses SMOTE, which was developed by Chawla et al. [40], and creates artificial examples for the minority class that are planted on similar features of the minority class. First, it looks for k-nearest neighbors (NNs) from minority examples. Then, furthermore, it selects random neighbors and creates an artificial sample at an arbitrarily chosen point between the two samples. For the second step, this algorithm employs ENN, which uses three nearest neighbors to edit misclassified samples, and then applies the single nearest neighbor rule to make decisions [41].

Let us assume that X_i is a set of minority class $X_i \in$Xminority; then, SMOTE selects k as its nearest neighbors Kx_i. Figure 3A illustrates an example of three nearest neighbors of X_i that are connected by a line with a set of minority class X_i. First, SMOTE creates a new example M, which belongs to X_i, by randomly selecting element N from Kx_i. The feature vector of new example M will be the sum of the feature vector of Xi and the value that

can be obtained by multiplying the difference between Xi, M, and random value β, whose value varies between 0 and 1.

$$M = X_i + (N - X_i)\beta \quad (1)$$

where N is an element from Kx_i such that $N \epsilon X_{minority}$. The newly generated example is a point between the line segment of Xi and a randomly selected point of N as Xi ϵKx_i. Figure 3B illustrates the SMOTE with a toy example in which a new example of M is created between the lines of X_i and N. After that, it applies ENN to remove the example from the dataset. ENN removes samples that differ from two other examples in the three nearest neighbors. Figure 3C illustrates ENN working with an example. Before applying SMOTE-ENN, the class distribution for city hotel for the majority and minority classes was 46,228 (58.27%) and 33,102 (41.73%), respectively; after applying this method, these values became 31,198 (55.70%) and 24,803 (44.29%). For the resort hotel, these values were 28,938 (74.24%) for majority class and 11,122 (27.76%) for minority class; after SMOTE-ENN, these values became 20,029 (55.54%) and 16,029 (44.45%) for majority and minority class. Figure 4 shows the SMOTE-ENN, and Figure 5 shows the flow diagram of the SMOTE-ENN.

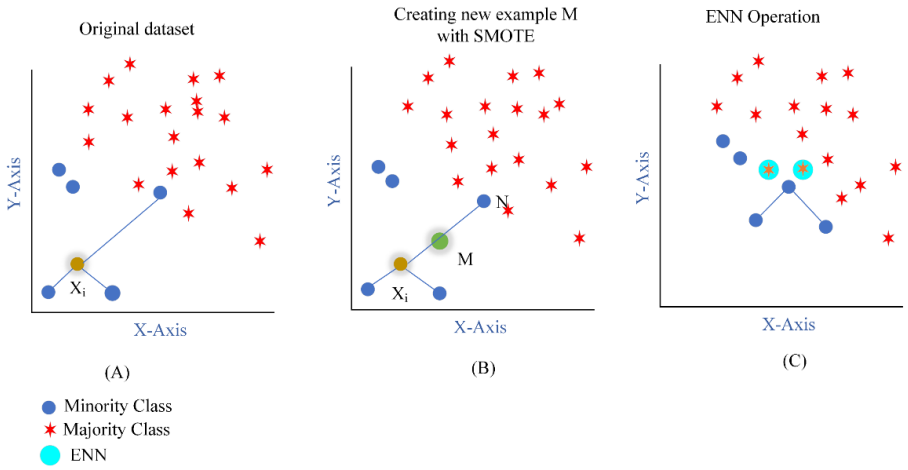

Figure 3. Illustration of SMOTE-ENN with a toy example [39]. (**A**) represent the dataset before applying SMOTE-ENN. (**B**) represent the SMOTE operation where new examples for minority class are generated and (**C**) represent the ENN operation in which unnecessary examples are removed.

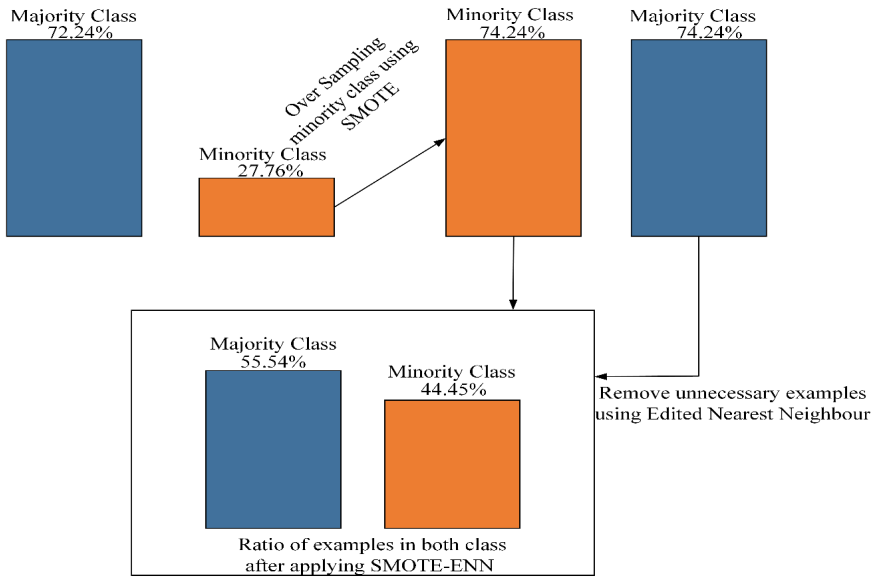

Figure 4. Illustration of SMOTE-ENN for Resort hotel (H1) [39].

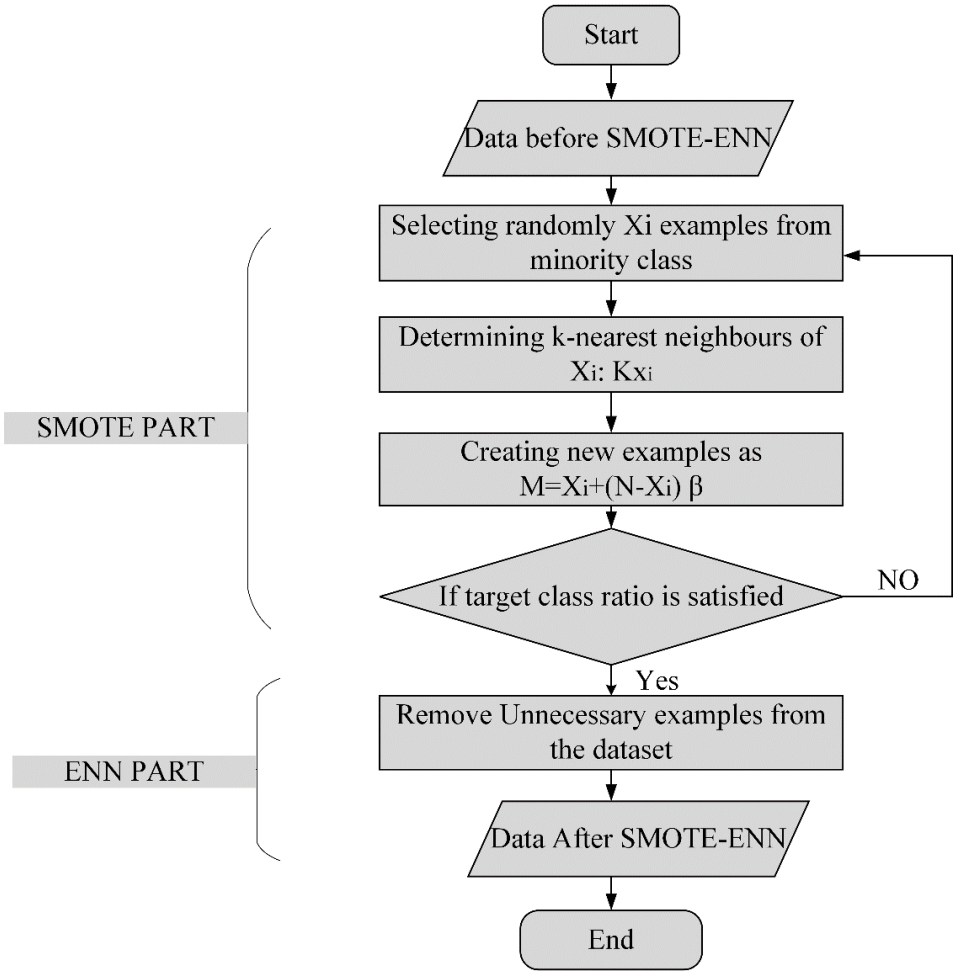

Figure 5. Flow diagram of SMOTE-ENN [39].

4. Modelling and Performance Evaluation

At the end of the SMOTE-ENN step, we trained a random forest classifier. Since all features had a diverse structure of importance or significance and weights per hotel (lodging), a particular model had to be developed for every hotel. As distinctive algorithms show distinctive outcomes, new models were created utilizing diverse classification methods; this was performed after selecting the ones that showed better execution indicators. As the name "IsCanceled" within the dataset could take two values (0: no; 1: yes), the adherents of two-class simple classification methods were chosen: logistic regression (LR), decision tree (DT), AdaBoost (AB), gradient boosting (GD) and random forest (RF).

All approaches were executed in Python 3.7 and the experiment was completed on a Windows 10 machine with a 16 GB RAM, 4 GB NVDIA GTX 1650Ti graphic card and a core i7 processor. In addition, SMOTE and SMOTE-ENN were executed by the imbalanced-learn bundle [42,43] and LR, DT, AB, GD, and RF in the Scikit-learn bundle [44]. The imbalanced-learn bundle is a free-source from the Python library that comprises many techniques for managing the issue of class imbalance, while the Scikit-learn bundle is a free machine learning library for the Python language.

To show the viability of our approach, we examined the exhibition among the stand-alone standard machine learning methods, the standard ML method with SMOTE, and the standard ML method with SMOTE-ENN. We used a standard method to predict hotel cancelation directly from the data, i.e., in those methods, we did not apply any resample methods prior to sending the data to the classifiers. For the second group of methods, we applied the oversampling method (SMOTE) prior to sending the data to the same classifiers. For the third group of methods, we applied a hybrid of under-sampling and

oversampling methods (SMOTE-ENN) prior to sending the same set of classifiers to access the performance of the classifier after the addition of class imbalance methods to adjust for class distribution. Additionally, this study utilized 10-fold cross-validation with a diverse arrangement of folds for each execution to achieve average performance. When using 10-fold cross validation, we utilized the GridSearchCV function in Scikit-learn that allowed us to choose the cross-validation scheme according to our needs; in this study, we used 10-fold cross-validation. Following this, we utilized GridSearchCV in the Scikit-learn bundle [44] to tune the parameters of RF.

We used different classification metrices to assess the performance of the proposed strategy on test data. Accuracy, precision, recall, AUC-ROC curve, AUC score, F1 Score, and G-mean were included to access the performance of the test data [45]. We also included a precision–recall (PR) curve, since some studies suggested that the ROC with an imbalanced dataset may well be tricky and lead to incorrected interpretations regarding the method's performance [46]. The reason behind this unusual behavior is because ROC and PR are diverse, since the latter targets the minority class, while ROC encompasses both classes. The precision–recall–auc (PR-AUC) score used to access the model's performance using a single digit [47]. We compared our results with the standard random forest and random forest with SMOTE, and concluded that the addition of SMOTE-ENN before the classifier increased random forest classifier performance while addressing the class imbalance problem in relation to hotel cancelation predictions. We selected different values for the random forest classifier, such as criterain: {'Entropy','Gini'}, Max-features: {'log2','Auto'}, Min-samples_leaf: {1, 2, 3, 4, 5}, Min_samples_split: {4, 5, 6, 7, 8}, and N-estimators: {100, 150, 200, 250, 300, 350, 400, 450}. All these values were passed as parameters inside the GridSearchCV function that was fitted 8000 times on the dataset to find optimal parameters for the random forest classifier. Optimal parameters for the classifier were achieved through a grid search. Table 2 shows the list of parameters of the random forest classifier.

Table 2. Optimal parameters for random forest classifier.

Optimal Parameter	Hotel H1	Hotel H2
Criterion	Entropy	Gini
Max_features	Log2	Auto
Min_samples_leaf	1	1
Min_samples_split	4	4
N_estimators	100	200

We assessed the performance of the classification model using the number of counts from the dataset that were correctly and incorrectly classified by the model. The counts are arranged in a square table recognized as a confusion matrix. There, "true positive" indicates that the classifier predicted values as true, and they were true in reality. Meanwhile, "false positive" indicates that the classifier predicted values were true, but they were false. "False negative" indicates that the classifier predicted values were negative, but they were true; "true negative" indicates that the classifier predicted values as negative, which they were.

AUC-ROC curve: Receiver operator characteristic (ROC) is a widely used performance metric in binary classification [48]. It plots true positive rates against false positive rates at different thresholds and separate signals against noise. Area under curve (AUC) measures the separability of a classification model for binary classification, and it also uses the ROC curve as a summary. There are other metrics that are important for calculating the AUC-ROC curve.

True Negative Rate: This recognizes to what extent the negative class accurately classified as negative is in fact negative.

$$\text{Specificity/TrueNegativeRate} = \frac{TrueNegative}{TrueNegative + FalsePositive} \quad (2)$$

False Positive rate: This identifies what proportion of the negative class is incorrectly classified as positive with respect to all negative classes.

$$\text{FalsePositiveRate} = \frac{FalsePositive}{TrueNegative + FalsePositive} \quad (3)$$

False Negative Rate: This distinguishes to what extent the positive class is inaccurately classified as a negative class by the classifier.

$$\text{FalseNegativeRate} = \frac{FalseNegative}{TruePositive + FalseNegative} \quad (4)$$

Figure 6 shows the ROC curve (Figure 6A,C) and the precision–recall curve (Figure 6B,D) for the H1 and H2 hotels. We can observe from the figures that, after addressing the imbalance problem, the performance of the classifier improves significantly. In addition to this, we found out that, even for the H2 hotel, which was not imbalanced by much, it was still able to perform better after applying the SMOTE-ENN method before feeding the data into the classifier. The H1 hotel was initially highly imbalanced; however, accuracy increased to a certain extent.

To assess the performance of different classifiers, we used the data from [24] as a case study in this research. We also reported the results of SMOTE with classifiers to give a better picture when it comes to applying SMOTE-ENN.

The results of SMOTE-ENN were promising. For both hotels, the lowest accuracy was 86.3%, which was achieved in the HI hotel with logistic regression, while random forest achieved more than 95% accuracy in both the hotels. All methods registered better accuracy compared to the standard and standard + SMOTE classifiers, except for LR+SMOTE, which received slightly better accuracy compared to LR + SMOTE-ENN. If we take AUC as an assessment measure, this is even better in all standard + SMOTE-ENN methods, as they registered better results compared to standard and standard + SMOTE classifiers. In terms of performance, RF + SMOTE-ENN was the most accurate algorithm. In terms of precision and recall, LR + SMOTE-ENN beat all other algorithms, including the standard and standard +SMOTE classifiers. For F1 Score and PR-AUC, RF + SMOTE-ENN turned out to be the best among all algorithms. In the case of G-mean, which is a multiplication of sensitivity and specificity, the classifier performance values were between 0 and 1. A value closer to 1 showed a better classifier, and RF + SMOTE-ENN achieved the best values of 95% and 96.3% for hotels H1 and H2, respectively.

Another significant measure is the count of false positives rate. A false positive rate is important in the event of a hotel taking action against a booking classified as "going to be canceled". In such cases, the model that generates the smallest number of false predictions is beneficial for a hotel, as such an establishment would need to spend fewer resources on bookings that are yet to be canceled. If such important criteria are taken into account, RF+SMOTE-ENN should be chosen for hotel cancelation predictions, as this algorithm presents the smallest number of false predictions among all algorithms.

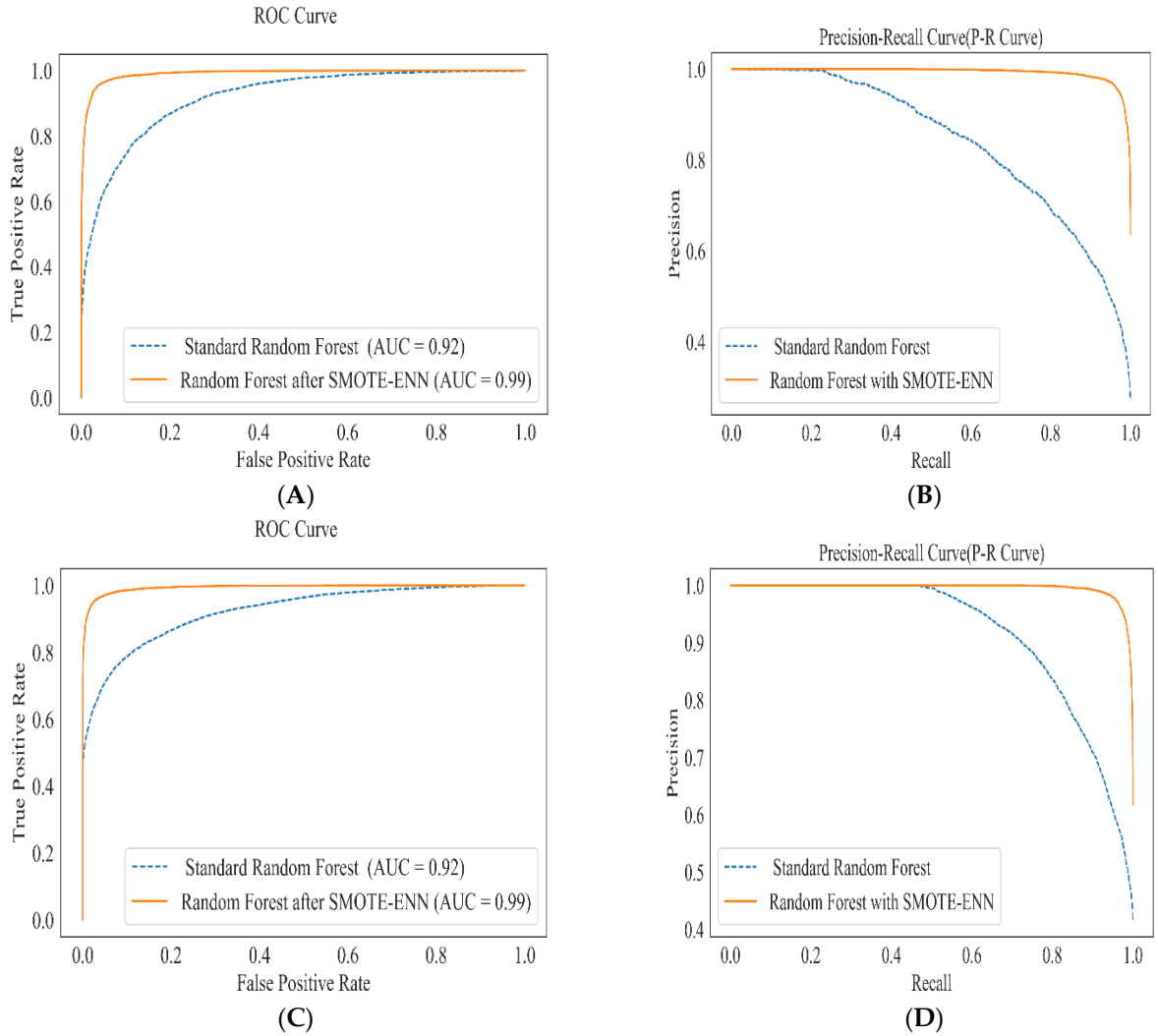

Figure 6. ROC-AUC and PR-curve analysis of H1 and H2 hotels.

For hotels to increase their revenue and make important decisions regarding their allocation of rooms, it is important that they accurately predict which customers might cancel their bookings in advance. Since hotel cancelation problems normally suffer from class imbalance issues, it is equally important to address this issue before applying any classifier for prediction, so that a model does not show bias toward the majority class [33]. Our inclusion of SMOTE-ENN in case of the hotel cancelation problem could benefit the hospitality industry if preexisting datasets are suffering from problems related to class imbalances. Gustavo Batista et al. investigated numerous combinations of oversampling and under-sampling strategies compared to currently utilized strategies [28]. Ultimately, the researchers noted that ENN was more effective at down sampling the majority class than the methods included in their study. They applied their strategy by expelling samples from both the majority and minority classes. Hence, any sample that was misclassified by its three closest neighbors was eliminated from the preparing set, which makes class distribution better for both classes and helps the classifier in its predictions compared to the SMOTE method itself. Tables 3 and 4 show the results of different classifier performances, and we can observe from the table that standard+ SMOTE-ENN improved performance compared to the standard and standard +SMOTE classifiers. Among all the classifiers, random forest achieved the best results. From all the results, we can observe that SMOTE-ENN is able to enhance the prediction performance of classifiers by a significant amount.

Table 3. Results for H1 hotel.

Method	TNR	FPR	FNR	Accuracy	Precision	Recall	F1 Score	AUC	PR-AUC	G-Mean
LR	75.82	24.17	15.71	82.60	93.44	84.26	88.61	73.77	75.44	71.10
LR + SMOTE	81.51	15.64	17.01	83.66	84.39	82.98	83.68	83.76	93.79	83.53
LR + SMOTE-ENN	86.34	13.65	15.80	**86.30**	**84.67**	**84.76**	**84.72**	**86.15**	**96.27**	**85.95**
DT	52.90	47.09	12.01	81.88	86.90	87.94	87.42	77.89	71.64	77.18
DT + SMOTE	88.99	14.01	12.71	86.61	85.48	87.28	86.37	86.60	89.88	86.45
DT + SMOTE-ENN	93.19	6.43	6.70	**93.41**	**91.99**	**93.21**	**92.61**	**93.28**	**95.56**	**93.04**
AB	76.97	23.02	15.47	83.04	93.76	84.52	88.90	74.29	77.67	71.70
AB + SMOTE	81.85	18.14	15.30	83.96	80.29	86.42	83.24	83.93	93.18	84.40
AB + SMOTE-ENN	86.32	13.67	11.77	**87.76**	**82.17**	**89.66**	**85.74**	**87.23**	**96.31**	**87.12**
GB	79.68	20.31	15.34	83.71	94.68	84.65	89.39	74.77	79.61	72.20
GB + SMOTE	83.49	16.50	11.48	85.37	82.22	87.51	84.80	85.34	90.00	85.35
GB + SMOTE-ENN	89.21	10.78	9.69	**89.68**	**96.24**	**90.30**	**88.22**	**89.35**	**96.96**	**89.46**
RF	78.69	21.30	11.67	85.97	92.78	88.43	90.55	80.42	83.78	79.40
RF + SMOTE	90.78	9.21	10.34	90.21	90.76	89.65	90.20	90.21	97.00	90.28
RF + SMOTE-ENN	94.95	4.54	4.49	**95.39**	**94.20**	**95.43**	**94.82**	**95.28**	**99.17**	**95.08**

Table 4. Results of the H2 hotel.

Method	TNR	FPR	FNR	Accuracy	Precision	Recall	F1 Score	AUC	PR-AUC	G-Mean
LR	85.75	14.24	20.42	81.58	92.07	79.57	85.37	79.47	86.16	78.48
LR + SMOTE	86.10	13.89	21.02	82.08	87.95	78.95	83.21	82.02	91.12	81.92
LR + SMOTE-ENN	91.93	8.06	17.51	**87.30**	**90.73**	**82.48**	**84.72**	**86.41**	**95.59**	**88.13**
DT	77.92	22.07	15.42	81.77	84.14	84.49	84.23	81.19	82.40	81.21
DT + SMOTE	83.03	16.96	15.45	83.78	83.06	84.54	83.80	83.79	87.58	83.80
DT + SMOTE-ENN	95.07	4.92	5.56	**94.79**	**93.81**	**94.43**	**94.12**	**94.69**	**96.50**	**94.57**
AB	85.53	14.76	20.07	81.70	91.68	79.92	85.40	79.90	87.51	78.80
AB + SMOTE	86.11	13.88	22.12	81.40	88.23	77.87	82.73	81.33	90.88	81.06
AB + SMOTE-ENN	92.60	7.39	17.56	**87.58**	**91.59**	**82.43**	**86.77**	**87.98**	**96.65**	**88.16**
GB	86.40	13.59	20.08	82.02	92.42	79.91	85.71	79.93	88.53	79.00
GB + SMOTE	88.33	11.66	21.68	82.49	90.35	78.31	83.90	82.42	91.93	81.98
GB + SMOTE-ENN	93.46	6.53	17.77	**89.04**	**92.46**	**84.40**	**88.24**	**89.38**	**97.25**	**89.73**
RF	85.57	14.20	14.85	85.39	90.81	85.14	87.88	84.32	91.72	84.15
RF + SMOTE	88.93	11.06	15.31	87.24	89.65	85.73	87.64	87.22	95.21	87.30
RF + SMOTE-ENN	96.98	3.01	4.87	**96.14**	**96.27**	**95.12**	**95.69**	**96.16**	**99.50**	**96.33**

Tables 3 and 4 present the true negative rate (TNR), false positive rate (FPR), and false negative rate (FPR) for both hotels. From the table, we can see that, for both hotels, RF+SMOTE-ENN achieved the highest TNR of almost 95% and 97%, which demonstrates that this classifier is able to accurately classify negative examples compared to other classifiers. Similarly, RF+SMOTE-ENN achieved the lowest false positive rates, 4.54 and 3.01, which shows that only 4.5% and 3% of the examples were misclassified as positive examples from all negative examples; this is an important measure regarding hotel booking cancelations. Furthermore, RF + SMOTE-ENN also achieved the lowest false negative rate,

which demonstrates the extent to which positive examples were misclassified as negative examples. RF + SMOTE-ENN achieved 4.49 and 4.87 FNR for both hotels, which are the smallest values among all classifiers. We presented a statistical test for the classifiers included in this study, and we used a 5 × 5 cv combined F-test to establish the statistical significance of all classifiers; this approach is recommended for the testing of a classifier in one dataset [10]. Tables 5 and 6 display the statistical significance of the different classifiers included in this study. We calculated the p-value of RF vs. every other classifier. All classifiers registered less value compared to a significance threshold of $\alpha = 0.05$, which shows that both classifier performances are not similar.

Table 5. Statistical significance of classifiers for hotel 1.

Methods	*p*-Value	F-Statistic
RF vs. LR	9.064×10^{-9}	134.339
RF vs. CLF	1.295×10^{-5}	158.175
RF vs. AB	2.400×10^{-5}	123.369
RF vs. GB	2.624×10^{-5}	119.012
RF + SMOTE vs. LR + SMOTE	1.364×10^{-7}	983.107
RF + SMOTE vs. CLF + SMOTE	1.024×10^{-7}	1102.694
RF + SMOTE vs. AB + SMOTE	$4.543. \times 10^{-7}$	607.249
RF + SMOTE vs. GB + SMOTE	1.052×10^{-7}	1091.102
RF + SMOTE-ENN vs. RF + SMOTE-ENN	3.518×10^{-7}	672.790
RF + SMOTE-ENN vs. CLF + SMOTE-ENN	3.495×10^{-5}	106.001
RF + SMOTE-ENN vs. AB + SMOTE-ENN	1.576×10^{-9}	368.736
RF + SMOTE-ENN vs. GB + SMOTE-ENN	9.064×10^{-9}	2910.776

Table 6. Statistical significance of classifiers for hotel 2.

Methods	*p*-Value	F-Statistic
RF vs. LR	6.128×10^{-6}	213.791
RF vs. CLF	8.897×10^{-7}	463.883
RF vs. AB	6.031×10^{-6}	215.159
RF vs. GB	6.372×10^{-6}	210.458
RF + SMOTE vs. LR + SMOTE	1.759×10^{-7}	888.049
RF + SMOTE vs. CLF + SMOTE	5.199×10^{-6}	228.393
RF + SMOTE vs. AB + SMOTE	4.823×10^{-6}	235.392
RF + SMOTE vs. GB + SMOTE	6.970×10^{-6}	203.003
RF + SMOTE-ENN vs. RF + SMOTE-ENN	1.573×10^{-7}	928.726
RF + SMOTE-ENN vs. CLF + SMOTE-ENN	1.26×10^{-4}	62.714
RF + SMOTE-ENN vs. AB + SMOTE-ENN	2.052×10^{-7}	834.984
RF + SMOTE-ENN vs. GB + SMOTE-ENN	1.249×10^{-6}	404.781

5. Conclusions

This study addressed the issue of class imbalance in hotel cancelation predictions. We introduced a SMOTE-ENN oversampling technique to address this issue. Our study shows that, after addressing this issue with SMOTE-ENN, the performance of a machine learning classifier increases significantly by introducing a combination of under-sampling and

oversampling methods (SMOTE-ENN). All models registered significant improvements compared to standard and standard + SMOTE classifiers. Among them, RF + SMOTE-ENN achieved the best results in all performance measures included in this study. The proposed methodology can address the issue of imbalance in datasets, and forecasting models can empower hotel supervisors to calculate their losses arising out of advanced booking cancelations and restrict issues related to overbooking (redistribution expenses, money or administration pay, and, especially significant today, social standing expenses).

6. Implications

Booking cancelation models may permit hotel supervisors to execute fewer lenient strategies without expanding their vulnerability. This could possibly result in more deals, as more flexible booking strategies create more clients.

Moreover, these classifiers can permit hotel supervisors to predict and prepare for bookings that are likely to be canceled. In addition, the hospitality industry can take advantage of this approach by using our proposed method to increase revenue by increasing classifier performances with more precise demand forecasting.

7. Limitations and Directions for Further Research

Despite achieving good results, there are a few limitations of this research. Since data for both hotels come from the same PMS database, questions should be asked regarding whether similar results could be achieved from other datasets. Moreover, if more hotels are included in the study, whether the proposed model would be able to achieve similar performance across the board is another important question. Consequently, future researchers can examine other potential class imbalance methods in addition to ours; some of these approaches may be more advanced and effective in examining hotel booking cancelations.

Author Contributions: Conceptualization, M.A. and A.A.; Data curation, M.F.A. and R.K.C.; Formal analysis, M.A., M.F.A. and J.-Z.W.; Investigation, M.A. and A.A.; Methodology, M.F.A. and R.K.C.; Project administration, A.A. and J.-Z.W.; Software, A.A. and R.K.C.; Supervision, M.A.; Writing—original draft, M.A. and M.F.A.; Writing—review and editing, A.A., J.-Z.W. and R.K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Taif University, Saudi Arabia, through the Taif University Researchers Supporting Project, under Grant "TURSP-2020/121" and the Ministry of Science and Technology, Taiwan (MOST108-2221-E-031-001-MY2; MOST110-2628-E-031-001) and the Center for Applied Artificial Intelligence Research, Soo-chow University, Taiwan (C-01).

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. Kimes, S.E.; Wirtz, J. Has revenue management become acceptable? Findings from an International study on the perceived fairness of rate fences. *J. Serv. Res.* **2003**, *6*, 125–135. [CrossRef]
2. Chiang, W.C.; Chen, J.C.H.; Xu, X. An overview of research on revenue management: Current issues and future research. *Int. J. Revenue Manag.* **2007**, *1*, 97. [CrossRef]
3. Mehrotra, R.; Ruttley, J. *Revenue Management*, 2nd ed.; American Hotel & Lodging Association (AHLA): Washington, DC, USA, 2006.
4. Talluri, K.T.; Van Ryzin, G.J. *The Theory and Practice of Revenue Management*; Kluwer Academic Publishers: Boston, MA, USA, 2004.
5. Smith, S.J.; Parsa, H.; Bujisic, M.; Van Der Rest, J.-P. Hotel Cancelation Policies, Distributive and Procedural Fairness, and Consumer Patronage: A Study of the Lodging Industry. *J. Travel Tour. Mark.* **2015**, *32*, 886–906. [CrossRef]
6. Chen, C.-C.; Schwartz, Z.; Vargas, P. The search for the best deal: How hotel cancellation policies affect the search and booking decisions of deal-seeking customers. *Int. J. Hosp. Manag.* **2011**, *30*, 129–135. [CrossRef]
7. Chen, C.-C.; Xie, K.L. Differentiation of cancellation policies in the U.S. hotel industry. *Int. J. Hosp. Manag.* **2013**, *34*, 66–72. [CrossRef]

8. Morales, D.R.; Wang, J. Forecasting cancellation rates for services booking revenue management using data mining. *Eur. J. Oper. Res.* **2010**, *202*, 554–562. [CrossRef]
9. Liu, P.H. Hotel demand/cancelation analysis and estimation of unconstrained demand using statistical methods. In *Revenue Management and Pricing: Case Studies and Applications*; Yeoman, I., McMahon-Beattie, U., Eds.; Cengage Learning EMEA: Bedford Row, London, UK, 2004; pp. 91–108.
10. Alpaydm, E. Combined 5× 2 cv F Test for Comparing Supervised Classification Learning Algorithms. *Neural Comput.* **1999**, *11*, 1885–1892. [CrossRef]
11. Noone, B.M.; Lee, C.H. Hotel overbooking: The effect of overcompensation on customers' reactions to denied service. *J. Hosp. Tour. Res.* **2010**, *35*, 334–357. [CrossRef]
12. Stanislav, I. *Hotel Revenue Management: From Theory to Practice*; Zangador: Varna, Bulgaria, 2014. Available online: https://ssrn.com/abstract=2447337 (accessed on 13 March 2021).
13. Hayes, D.K.; Miller, A.A. *Revenue Management for the Hospitality Industry*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2011.
14. Freisleben, B.; Gleichmann, G. Controlling airline seat allocations with neural networks. In Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences, Wailea, HI, USA, 8 January 1993.
15. Garrow, L.; Ferguson, M. Revenue management and the analytics explosion: Perspectives from industry experts. *J. Revenue Pricing Manag.* **2008**, *7*, 219–229. [CrossRef]
16. Hueglin, C.; Vannotti, F. Data mining techniques to improve forecast accuracy in airline business. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001. [CrossRef]
17. Lemke, C. Combinations of Time Series Forecasts: When and Why Are They Beneficial? Bournemouth University, 2010. Available online: http://dec.bournemouth.ac.uk/staff/bgabrys/publications/C_Lemke_PhD_thesis.pdf (accessed on 19 March 2021).
18. Subramanian, J.; Stidham, S., Jr.; Lautenbacher, C.J. Airline Yield Management with Overbooking, Cancellations, and No-Shows. *Transp. Sci.* **1999**, *33*, 147–167. [CrossRef]
19. Gil Yoon, M.; Lee, H.Y.; Song, Y.S. Linear approximation approach for a stochastic seat allocation problem with cancellation & refund policy in airlines. *J. Air Transp. Manag.* **2012**, *23*, 41–46. [CrossRef]
20. Schwartz, Z.; Uysal, M.; Webb, T.; Altin, M. Hotel daily occupancy forecasting with competitive sets: A recursive algorithm. *Int. J. Contemp. Hosp. Manag.* **2016**, *28*, 267–285. [CrossRef]
21. Caicedo-Torres, W.; Payares, F. A machine learning model for occupancy rates and demand forecasting in the hospitality industry. Presented at the Ibero-American Conference on Artificial Intelligence, San José, Costa Rica, 23–25 November 2016; Springer: Cham, Switzerland, 2016; pp. 201–211.
22. Antonio, N.; de Almeida, A.; Nunes, L. Using data science to predict hotel booking cancelations. In *Handbook of Research on Holistic Optimization Techniques in the Hospitality, Tourism, and Travel Industry*; Vasant, P., Kalaivanthan, M., Eds.; Business Science Reference: Hershey, PA, USA, 2017; pp. 141–167.
23. Huang, H.-C.; Chang, A.Y.; Ho, C.-C. Using artificial neural networks to establish a customer-cancelation prediction model. *Prz. Elektrotech.* **2013**, *89*, 178–180.
24. Antonio, N.; De Almeida, A.; Nunes, L. Predicting hotel booking cancellations to decrease uncertainty and increase revenue. *Tour. Manag. Stud.* **2017**, *13*, 25–39. [CrossRef]
25. Antonio, N.; De Almeida, A.; Nunes, L. An Automated Machine Learning Based Decision Support System to Predict Hotel Booking Cancellations. *Data Sci. J.* **2019**, *18*, 1–20. [CrossRef]
26. Antonio, N. Predictive models for hotel booking cancellation: A semi-automated analysis of the literature. *Tour. Manag. Stud.* **2019**, *15*, 7–21. [CrossRef]
27. Leevy, J.; Khoshgoftaar, T.M.; Bauder, R.A.; Seliya, N. A survey on addressing high-class imbalance in big data. *J. Big Data* **2018**, *5*, 42. [CrossRef]
28. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behaviour of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
29. Le, T.; Vo, M.T.; Vo, B.; Lee, M.Y.; Baik, S.W. A Hybrid Approach Using Oversampling Technique and Cost-Sensitive Learning for Bankruptcy Prediction. *Complexity* **2019**, *2019*, 8460934. [CrossRef]
30. Schmidt, J.; Marques, M.R.G.; Botti, S.; Marques, M.A.L. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **2019**, *5*, 1–36. [CrossRef]
31. Dimiduk, D.M.; Holm, E.A.; Niezgoda, S.R. Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. *Integrating Mater. Manuf. Innov.* **2018**, *7*, 157–172. [CrossRef]
32. Attaran, M.; Deb, P. Machine learning: The new 'big thing' for competitive advantage. *Int. J. Knowl. Eng. Data Min.* **2018**, *5*, 277–305. [CrossRef]
33. Patel, H.; Purvi, P. Study and Analysis of Decision Tree Based Classification Algorithms. *Int. J. Comput. Sci. Eng.* **2018**, *6*, 74–78. [CrossRef]
34. Gil Yoon, M.; Lee, H.Y.; Song, Y.S. Dynamic pricing & capacity assignment problem with cancellation and mark-up policies in airlines. *Asia Pac. Manag. Rev.* **2017**, *22*, 97–103. [CrossRef]
35. Oussous, A.; Benjelloun, F.-Z.; Lahcen, A.A.; Belfkih, S. Big Data technologies: A survey. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *30*, 431–448. [CrossRef]

36. Feng, F.; Li, K.-C.; Shen, J.; Zhou, Q.; Yang, X. Using Cost-Sensitive Learning and Feature Selection Algorithms to Improve the Performance of Imbalanced Classification. *IEEE Access* **2020**, *8*, 69979–69996. [CrossRef]
37. Chen, R.-C.; Dewi, C.; Huang, S.-W.; Caraka, R.E. Selecting critical features for data classification based on machine learning methods. *J. Big Data* **2020**, *7*, 1–26. [CrossRef]
38. Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78–87. [CrossRef]
39. Flath, C.M.; Stein, N. Towards a data science toolbox for industrial analytics applications. *Comput. Ind.* **2018**, *94*, 16–25. [CrossRef]
40. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
41. Wilson, D.L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Trans. Syst. Man Cybern.* **1972**, *SMC-2*, 408–421. [CrossRef]
42. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **2017**, *18*, 559–563.
43. Antonio, N.; de Almeida, A.M.; Nunes, L. Big Data in Hotel Revenue Management: Exploring Cancellation Drivers to Gain Insights into Booking Cancellation Behavior. *Cornell Hosp. Q.* **2019**, *60*, 298–319. [CrossRef]
44. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
45. Natarajan, N.; Koyejo, O.; Ravikumar, P.; Dhillon, I. Consistent Binary Classification with Generalized Performance Metrics. In Proceedings of the Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 2744–2752.
46. Saito, T.; Rehmsmeier, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef]
47. Beger, A. Precision-Recall Curves. 2016. Available online: https://ssrn.com/abstract=2765419 (accessed on 13 March 2021).
48. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]