Predicting Hotel Reservation Cancellations

ORIE 5741 Paper

Nimra Shakoor (ns924), Preeti Uppuluri (pnu3), Ashley Yu (yy346)

Introduction

Booking cancellations result in losses of profit for hotels. When a customer cancels a booking, there is a cost to the hotel if they are not able to find someone else to book the room in time. Moreover, the percentage of booking cancellations within total bookings has been increasing over time since the ease of booking in recent years means that people can book rooms without being sure of their plans [1]. Knowing the likelihood of a hotel booking being canceled can aid hotels with this issue since it might help them decide how to take action in advance, such as through pricing, advertising, or overbooking if they are expecting a room cancellation.

We are pursuing the following whether we can predict the likelihood of a hotel booking being canceled. We will explore this question by analyzing a dataset of hotel reservation records from 2015 to 2018. Using data from this timeframe will ensure we can develop a model for a future that is more similar to the pre-pandemic era during which traveling and hotel booking were more frequent. The outcome variable is the booking status which would be either "Canceled" or "Not Canceled."

This dataset is well-suited for predicting hotel booking cancellations due to several key factors. First, the fields in the dataset are well-labeled and clearly defined, making it easy to understand and work with. Second, the dataset includes a binary classification of each booking as either canceled or not canceled, providing a clear target variable for predictive modeling. Third, the dataset contains a diverse range of features that are likely to be relevant to predicting cancellations, such as room type reserved, repeated guests, number of adults/children, booking channel, and previous cancellations. All these features would be commonly available in most hotels, as a result, it would be easier to recreate this dataset and train and test new models. Finally, the dataset is relatively large, with over 20,000 records, and includes data from two different hotels, allowing for robust analysis of cancellation patterns across different contexts. Altogether, these factors make the Hotel Booking Demand Dataset a strong choice for predicting hotel booking cancellations.

Dataset

The dataset we will be using was collected by Antonio et al. in 2019 from hotels in Portugal [2]. The first hotel, H1, is located in Algarve, and the other, H2, is located in Lisbon. Each observation in the dataset represents a hotel booking and the timeframe ranges from July 1, 2015, to August 31, 2017 [3]. Using data from this timeframe will ensure we can develop a model for a future that is more similar to the pre-pandemic era during which traveling and hotel booking were more frequent. We originally planned to use another dataset, however, the description did not indicate whether or not data was real or simulated. Since we wanted to use real-world data, we opted to switch our dataset to one with similar features and data from real hotels. Overall, there are 32 columns in the dataset including whether or not the reservation was canceled, the lead time, the arrival date, the reserved room type, the deposit type, and the total number of special guests. A more detailed description of each column is included in the Appendix. The dataset had 119,390 records in total, 40,060 from H1 and 79,330 from H2. However, since we did not have the computational ability to run models with such as large dataset, we chose to work with 20% of the data or 23,878 records.

Methodology

Exploratory Data Analysis & Data Processing

EDA

We began by performing some exploratory data analysis. Specifically, we looked at the proportion of cancellations in the entire data, as well as cancellations over time.

From Figure 1, we can see the proportion of cancellations within this data set is about 37 percent. Therefore, this data set is only mildly unbalanced and we do not need to incorporate class weights but rather treat this as a normal classification problem.

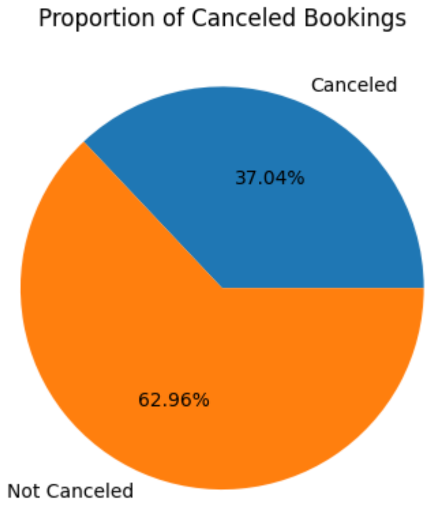


Figure 1: Percentage of cancellations in dataset

In Figure 2 we can see the cancellations per month over the years. As shown, the ratio of canceled to total bookings appears to be mostly consistent throughout the year, with peak cancellations occurring in May and October. Since cancellations are pretty consistent throughout the year, and we are including months as a feature, using a classification model is appropriate as opposed to a time series, for example.

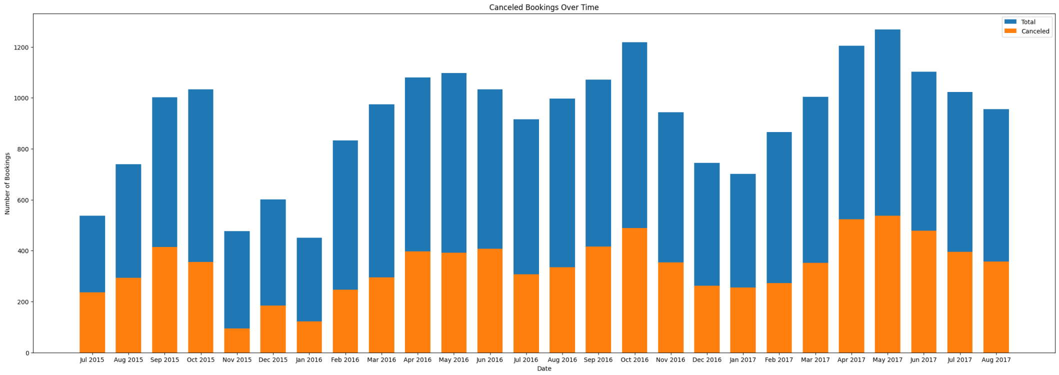


Figure 2: Cancelled Bookings Over Time

Processing

After sampling 20% of our entire dataset, we began processing the data by removing null values. In total, we had four columns with null values: `children`, `country`, `agent`, and `company`. `children` had one record, while `country` 103. However, since this only comprised about 3% of the data, we dropped those records. `agent` and `company`, which consist of numerical ID data, had many more, therefore we replaced all null IDs with zero — a value not already being used.

The next step was to perform a correlation analysis of the numerical columns (see Appendix) in our dataset. The objective was to determine which features were highly correlated so we can reduce the dimension of our data. However, as seen in Figure 3, none of the features had a very strong correlation, i.e. a correlation greater than 0.5 or less than -0.5, with the other features. Thus, we did not remove any numerical features.

The final step of our preprocessing was encoding our categorical variables. As shown in the Appendix, we divided our categorical features into ordinal variables, those with an inherent order, and nominal variables, having no order. Most of the ordinal columns were already encoded as numbers, such as the arrival year, the arrival week number, and the company ID. Only the arrival month had to be converted from the month name to its corresponding month number. The nominal columns were then one-hot encoded using the `pandas get_dummies` method. This resulted in a final dataset with 23,775 records, 202 features, and 1 outcome variable (`is_canceled`).

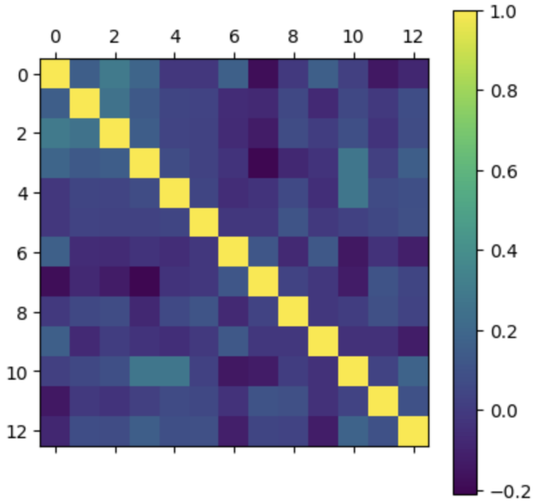


Figure 3: Correlation among numerical features in dataset

Train, Test, Validation, & Evaluation

After processing the data, we will randomly select 70% of the data points as training data and the remaining 30% as testing data. Then, with the training dataset, we will run a randomized grid search for different combinations of hyperparameters for each of our models. The optimal combination will have the highest mean accuracy on the validation sets from 3-fold cross-validation. More details on the hyperparameters for each model will be included in the **Algorithms** section. The final performance of the model will be measured by accuracy, precision, recall, and F1 score. The F1 score is generally a better measure of accuracy when classes are imbalanced or when false positives or false negatives are crucial. We will also look at the false negative rate, which indicates how often the model predicts someone will not cancel their reservation when, in fact, they do, with no cancelations being positive and cancelations being negative.

Algorithms

SVM

The first method we employ on this dataset is a Support Vector Machine (SVM). SVM is a supervised machine learning algorithm that can be used for classification or regression. It finds the optimal hyperplane that classifies a set of data points by maximizing the margin distance from each set of points to the hyperplane. In this case, we use SVM because it can help us classify whether a hotel booking will be canceled. When we run SVM we use the combinations of hyperparameters listed in Table 1.

Hyperparameter	Description	Values
`C`	Regularization parameter	[0.01, 0.1, 1, 10]
`kernel`	Kernel type	[poly, rbf]

Table 1: Hyperparameter values for random forest grid search

Random Forests

The second algorithm we will be using is a random forest classifier. It is an ensemble classifier that utilizes bagging and random feature selection to create uncorrelated datasets, on which decision trees are trained and then combined. The final outcome of a classification algorithm is determined by a majority vote. The main advantages of a random forest classifier are that it does not require scaling the data and it performs well on high dimensional data since it does not use a distance metric [4]. Thus, we believe our dataset is well-suited for this model since we have 202 features after encoding the nominal input variables. We will be running a grid search on the hyperparameter values listed in Table 2.

Hyperparameter	Description	Values
`n_estimators`	The number of trees in the forest	[200, 1000, 2000]
`max_depth`	The maximum depth of the tree	[10, 50, 100, None]

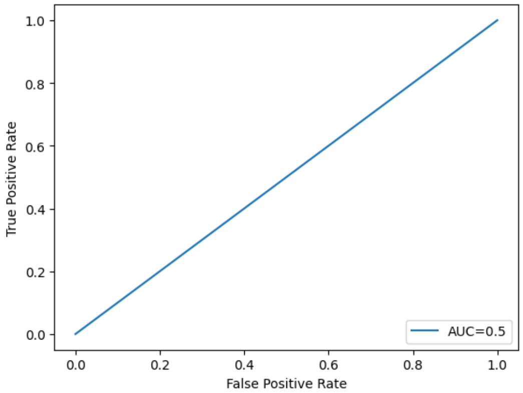
Table 2: Hyperparameter values for random forest grid search

Logistic Regression

The third algorithm we will be using is logistic regression. Logistic regression is a statistical model that models the probability of an event happening by having the log odds for the event be a linear combination of one or more independent variables. The parameters for logistic regression are most commonly estimated by maximum-likelihood estimation. We use logistic regression here because the dependent variable "is_canceled" that we are trying to predict is a binary variable, and our sample size is large enough to represent values across all response categories [5]. We will be running a grid search on the hyper-parameter values listed in Table 3.

Hyperparameter	Description	Values
`C`	Regularization parameter	[0.01, 0.1, 1, 10]
`solver`	Optimization algorithm	['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga']

Table 3: Hyperparameter values for random forest grid search



(a) SVM AUC-ROC curve

Metric	Value
Accuracy	0.63
Precision	1
Recall	0.63
F1	0.78
False Negative Rate	1

(b) SVM metrics

Figure 4: SVM model performance details

Results

SVM

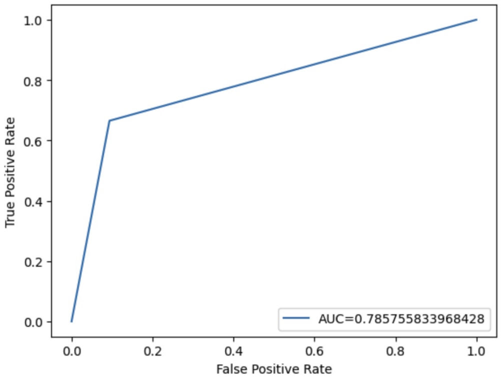
The results of the SVM, are shown in Table 4b. All combinations of the C and kernel hyperparameters yielded very similar cross-validations scores of about 0.63. While the precision is high, the false negative rate is 1 and the model did not perform well in terms of the other results either. In particular, the accuracy is quite low. Moreover, in Figure 4a we can see the AUC is 0.5, meaning the model is not able to distinguish well between the two classes. The model is not predicting any cancellations, therefore we can say it is not very accurate and has no predictive value.

Logistic Regression

The mean cross-validation score of running logistic regression on our dataset over multiple hyperparameter combinations was 0.762. This optimal result was obtained by using 'newton-cholesky' as a solver and 10 as the C value, which is significantly better than SVM. In Table 5b we can see the performance of the logistic regression model trained with these optimal parameters. In Figure 5a we can see the AUC-ROC curve for logistic regression. Based on the AUC value of about 0.79, we can say the logistic regression is relatively good at distinguishing between the two classes, compared to SVM, but the false negative rate is not as good as that of Random Forest.

Random Forest

The best-performing hyperparameter combination for random forests was 2000 estimators and a max depth of 50, which had a mean cross-validation accuracy score of 0.59. The performance of the random forest model trained with these optimal parameters is shown in Table 6.



(a) Logistic Regression AUC-ROC curve

Metric	Value
Accuracy	0.82
Precision	0.91
Recall	0.82
F1	0.86
False Negative Rate	0.18

(b) Results for logistic regression model

Figure 5: Logistic regression model performance details

Metric	Value
Accuracy	0.87
Precision	0.91
Recall	0.89
F1	0.90
False Negative Rate	0.11

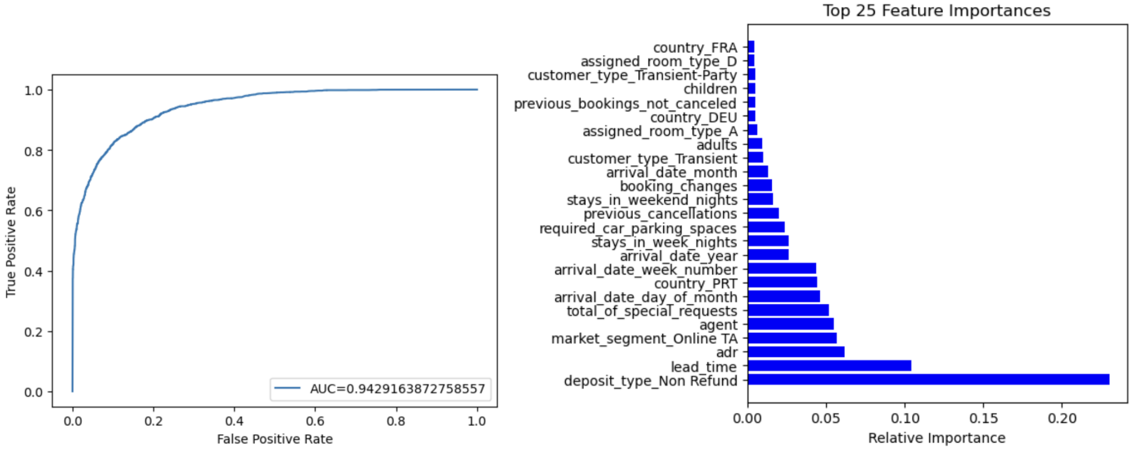
Figure 6: Results for random forest model

From the results, we see that precision, recall, accuracy, and F1 are relatively high and slightly higher than the logistic regression model. The false negative rate is also lower compared to logistic regression. From the height of the AUC-ROC curve and the AUC value of 0.94 in Figure 7a, we can see the random forest is much better at distinguishing the two classes. From Figure 7b we see the top 25 most important features the model utilizes. The model suggests deposit type is most important for determining whether a customer will cancel. This makes sense since if the deposit is refundable a customer will likely not hesitate to cancel. The lead time and price, or average daily rate (ADR), are also important factors.

Conclusion

Overall, our best-performing model is random forest, which has the highest accuracy, precision, recall, and F1 scores, and the lowest false negative rate. Moreover, the random forest has a high AUC value, allowing it to better distinguish between the two classes. Therefore, we have chosen it as the best-performing model.

We are confident in the reliability of our results since we utilized a high-quality dataset from a



(a) Random forest AUC-ROC curve

(b) Random forest feature importance

Figure 7: Random forest model performance details

reliable source and we employed multiple metrics and validation techniques during model evaluation to confirm the robustness of our results. Furthermore, our model is highly interpretable, with the most important factors in predicting hotel cancellation rates being logical and easy to understand. Based on these factors, we believe our models could be used in production to inform hotel policies regarding booking cancellations for the two hotels in Portugal that the dataset was collected from. However, before the model is implemented in practice, additional testing will need to be done to test whether the model can still perform well with more recent data and whether or not it can generalize to other hotels.

We recognize the potential for our model to become a “Weapon of Math Destruction” if new features are continuously added to make it less interpretable and if it is used inappropriately. Particularly, if it is being used to deny certain people or increase the price of reservations. This also brings to question the idea of fairness and whether or not it is fair for a business to deny reservations. To prevent harm, it is important to ensure that our model is being used ethically and responsibly and that the results are not being used to discriminate against guests or make decisions that could have negative consequences. Therefore, we recommend that the model be used to understand the factors behind cancellations and inform policy, rather than predicting whether individual customers will cancel. For example, one policy decision could be to no longer have refundable deposits since the deposit type seems to be a large factor in whether or not a customer cancels.

References

[1] P. Delgado, "Cancellations shooting up: Implications, costs and how to reduce them," May 2016. [Online]. Available: https://www.mirai.com/blog/cancellations-shooting-up-implications-costs-and-how-to-reduce-them/

[2] T. Mock and A. Bichat, "Hotel booking demand," 2019.

[3] N. Antonio, A. de Almeida, and L. Nunes, "Hotel booking demand datasets," *Data in Brief*, vol. 22, pp. 41–49, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340918315191

[4] IBM, "What is random forest?" 2021. [Online]. Available: https://www.ibm.com/topics/random-forest

[5] ——, "What is logistic regression?" 2022. [Online]. Available: https://www.ibm.com/topics/logistic-regression

Appendix

Dataset

Column Name	Description	Type
`hotel`	The type of hotel (H1 = Resort Hotel or H2 = City Hotel)	Categorical
`is_canceled`	Value indicating if the booking was canceled (1) or not (0)	Categorical
`lead_time`	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date	Numerical
`arrival_date_year`	Year of arrival date	Ordinal
`arrival_date_month`	Month of arrival date	Ordinal
`arrival_date_week_number`	Week number of year for arrival date	Ordinal
`arrival_date_day_of_month`	Day of arrival date	Ordinal
`stays_in_weekend_nights`	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel	Numerical
`stays_in_week_nights`	Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel	Numerical
`adults`	Number of adults	Numerical
`children`	Number of children	Numerical
`babies`	Number of babies	Numerical
`meal`	Type of meal booked. It has five different categories	Categorical
`country`	Country of origin. Categories are represented in the ISO 3155–3:2013 format	Categorical
`market_segment`	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”	Categorical

`distribution_channel`	Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”	Categorical
`is_repeated_guest`	Value indicating if the booking name was from a repeated guest (1) or not (0)	Categorical
`previous_cancellations`	Number of previous bookings that were cancelled by the customer prior to the current booking	Numerical
`previous_bookings_not_canceled`	Number of previous bookings not cancelled by the customer prior to the current booking	Numerical
`reserved_room_type`	Code of room type reserved. Code is presented instead of designation for anonymity reasons.	Categorical
`assigned_room_type`	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons.	Categorical
`booking_changes`	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS.	Numerical
`deposit_type`	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit – no deposit was made; Non-Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of the stay.	Categorical
`agent`	ID of the travel agency that made the booking	Ordinal

company	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons	Ordinal
days_in_waiting_list	Number of days the booking was in the waiting list before it was confirmed to the customer.	Numerical
customer_type	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it; Group – when the booking is associated to a group; Transient – when the booking is not part of a group or contract, and is not associated to other transient booking; Transient-party – when the booking is transient, but is associated to at least other transient booking.	Categorical
adr	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights.	Numerical
required_car_parking_spaces	Number of car parking spaces required by the customer.	Numerical
total_of_special_requests	Number of special requests made by the customer (e.g. twin bed or high floor).	Numerical

Table 4: Details about dataset columns

Group Member Contributions

Group Member Name	Contributions
Nimra Shakoor	Training and optimizing svm model Writing **Introduction** section Writing *EDA* section Writing *SVM* sections under **Algorithms** and **Results**
Preeti Uppuluri	Training and optimizing random forests model Writing **Dataset** section Writing *Processing* section Writing **Train, Test, Validation, & Evaluation** section Writing *Random Forests* sections under **Algorithms** and **Results**
Ashley Yu	Training and optimizing logistic regression model Writing *Logistic Regression* sections under **Algorithms** and **Results** Writing **Conclusion** section

Table 5: Group member contributions