UNIVERSITY OF CALIFORNIA, SANTA BARBARA

PSTAT 135/235: Group Project

Big Data Analytics of Hotel Bookings

Shuying Yu Philip Yoon Fei Xu Kenneth Liu

December 11, 2020

Abstract

This paper introduces the machine learning methods and big data approaches the group used to the predict whether hotel booking reservations will be canceled or not by customers using. Exploratory data analysis was conducted and necessary pre-processing steps were employed to prepare the data for analyses. Different classification models were built using logistic regression, random forest, and support vector machine models to make predictions. For each model, model accuracy, precision, recall (sensitivity), and other model evaluation metrics were collected for comparison. Different hyperparameters were also adjusted for sensitivity analysis. The results revealed that the best performing model was the logistic regression model with an intercept, as it retained high accuracy in model performance and retained good precision and recall scores. Future directions include using k-fold cross validation when training the models and tuning other hyperparameters and studying the weights of different variables.

Key words: big data, statistics, exploratory data analysis, classification, logistic regression, random forest, support vector machine, model evaluation, sensitivity analysis

Contents

1 Introduction

1.1 Dataset

Traveling pre-COVID-19 was both a leisure and business activity that supported the service and tourism industries, which include but are not limited to restaurants, small business owned shops, and hotels. One of the biggest travel expenses outside of airline tickets would be accommodation, and hotel bookings can be competitive depending on the time of the year and country people travel to. However, there may be instances where hotels may be canceled due to change of mind or change in travel restrictions due to lockdown instructed by the government (i.e., the pandemic)! In order to ensure hotels have enough rooms to accommodate current customers, they need to keep track of rooms that were booked and determine how likely customers will cancel their reservation.

The aim of the project is to predict whether a hotel booking will be canceled by the customer based on various factors, such as the type of hotel (city or resort), time between booking date and arrival date, number of children and adults per room, and so forth using machine learning methods and big data approaches.

1.2 Exploratory Data Analysis

The data includes 32 variables, which were a mix of categorical and numeric variables. Some descriptive statistics were taken from the numeric variables. These include the number of nights stayed during the weekend and weekday, as well as the number of days between booking and arrival date. The number of adults, children, and babies are also examined. The descriptions of some of the variables of interest are shown in Table 1.1 and in Table 1.2.

Table 1.1: Description of some the variables in the data set.

Variable Name	Description
`is_canceled`	Whether booking was canceled (1) or not (0)
`lead_time`	Days between booking and arrival date
`arrival_date_year`	Year of arrival date (2015 - 2017)
`arrival_date_month`	Month of arrival date (Jan - Dec)
`hotel_type`	City or resort hotel
`deposit_type`	No deposit, refundable, non-refundable

Table 1.2: Descriptive statistics for some numeric variables.

Variable Name	Minimum	Maximum	Mean	Std. Dev.
`adr`	-6.38	5400	101.83	50.54
`stays_in_weekend_nights`	0	19	0.93	0.99
`stays_in_week_nights`	0	50	2.50	1.91
`lead_time`	0	737	104.01	106.86

The variable `lead_time` ranges from 0 to 737 days, meaning that there are customers who book their hotel reservation well in advance (≥ 2 years). On average, customers would book their hotels about 3 months prior to arriving at the hotel. The variable `adr` is the average daily rate defined by dividing the sum of all lodging transactions by the total number of staying nights. The average daily rate was about 102 monetary units (price likely adjusted for each country). Average stay for weekday nights was higher than the average stay for weekend nights, which could make sense given that travelers may be staying for two weekday nights at the end of the week while only staying for one weekend night to maximize their budget. Figure 1.1 illustrates the frequency of days stayed at hotels.

On average, customers booking hotels do not bring babies or children (average around 0). One average, the bookings included 2 adults. Since the dataset contained variables such as `agent`, it could be inferred that the dataset represents adults who travel and stay at hotels for business or work, adults who are single couples, and adults who are on vacation with children.

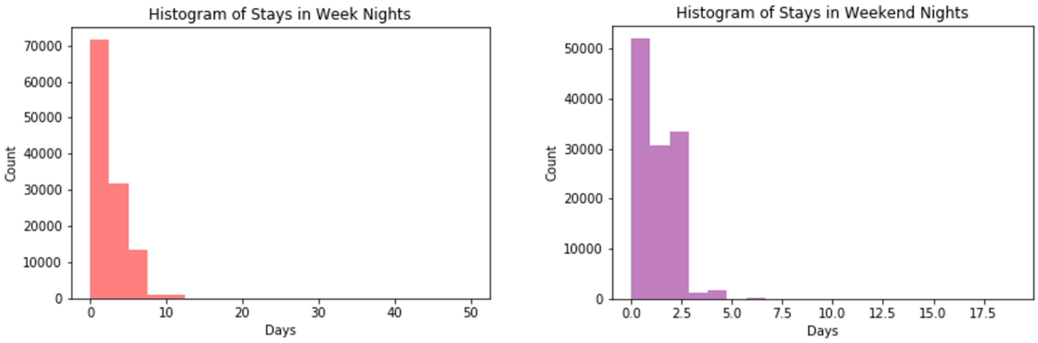

Figure 1.1: Histogram plots for the number of days booked for hotel stays.

Certain categorical variables are grouped together to study the difference in frequency of cancellations (Figure 1.2). The number of cancellations by hotel type (city versus resort) was first variable observed. City hotels tend to preserve bookings more than resort hotels, while also more likely to have customers cancel on their reservations than resort hotel. This may be due to city hotels being more represented in the dataset.

Another observational difference in hotels is whether they require deposits or not, and whether those deposits were refundable. Cancellations were more frequent when the bookings did not provide refundable deposits. When broken down by hotel type, the majority (about 10k more) of cancellations were for city hotels. Lastly, for the deposits that were refundable, the number of cancellations were fairly low. However, the original data set revealed that about 88% of hotels did not require a deposit to begin with.

Next, when looking into year or month of arrival date, the frequency of cancellations varied from year-to-year. However, the months with the lowest count of cancellations were November to February, which could indicate that this is around the holiday season when people would be less likely to cancel their trips and therefore their hotel reservations. The highest counts of cancellations were months spanning from May to August, which are the warmer months in the summer.

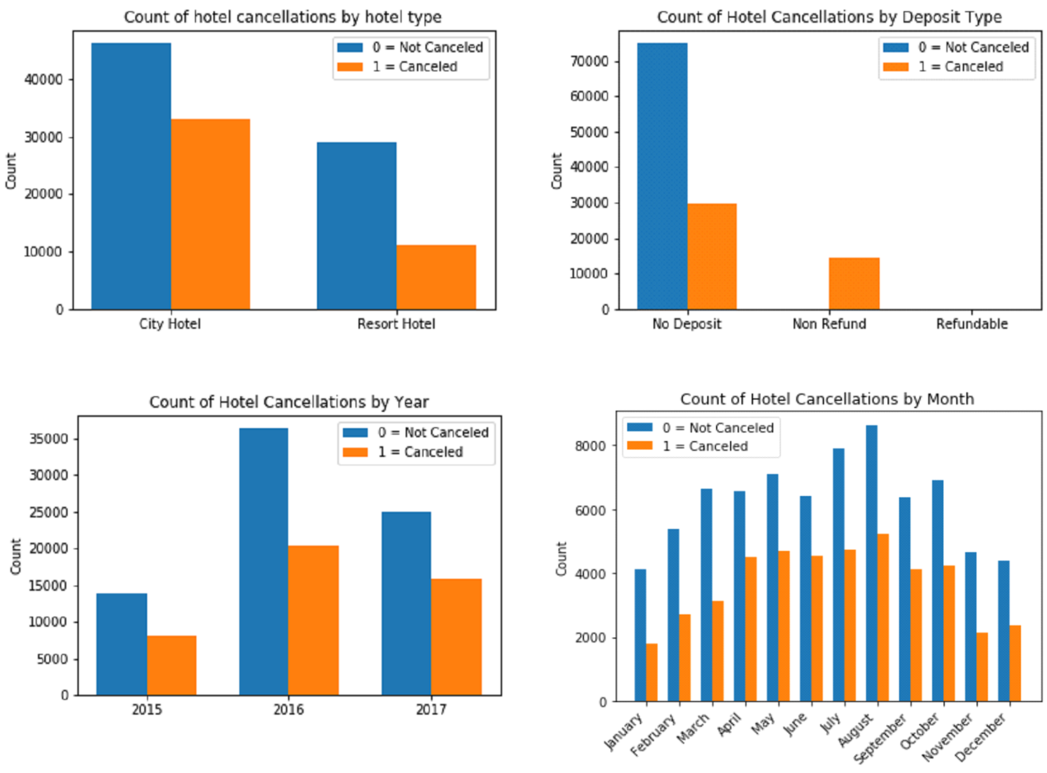

Figure 1.2: Bar plots for the frequency of whether hotel bookings were canceled for the variables hotel type (top left), deposit type (top right), arrival year (bottom left), and arrival month (bottom right).

2 Methods

2.1 Pre-processing

The ratio of data between the two classes for the response variable was about 6 to 4. Although the dataset was not even in both classes, the ratio is in an acceptable margin (i.e., not extreme like a ratio of 100 to 1, or even 10 to 1), so the class with more data was not downsampled.

After exploratory data analysis, the number of missing values the variables was assessed. String placeholders “NULL” and “N/A” were replaced with `None`, and the number of missing cells from each column were counted. Some variables had a high count of missing values. Several variables `agent`, `company`, and `country` were dropped because of the great amount of missing values. `arrival_date_week_number` and `reservation_status_date` were removed since arrival dates and the `lead_time` column were redundant. Variable `reservation_status` was also removed because it was 100% correlated with the labels.

Since the features are supposed to be numerical, the string columns were mapped to integers (e.g., ‘January’ in variable `arrival_date_month` to be a number ’00’) and all relevant categorical variables were converted to dummy variables using one-hot encoding. Additionally, continuous numerical variables were normalized using the L1 norm to standardize units. Then, the dataframe was converted into a resilient distributed dataset (RDD) and all features were combined to be left with two columns: label `is_canceled` and a dense vector of all features `features`. This was mapped again into a LabeledPoint object, and then split into 70/30: 70% reserved for training data and 30% reserved for testing data.

2.2 Building the Models

Classification models were expected to be used for this classification problem in predicting whether a hotel booking will be cancelled or not.

The simple model used was the logistic regression model. Since the response variable was a binary classification of 0 or 1 for non-cancel or cancel. The generalized linear model looks for a linear plane to separate the data into binary labels. The model was tested with and without intercepts, iterations (10, 50, 200), and both to assess how that changes the base accuracy of the simple model.

The random forest model was chosen in the project because they make up an ensemble of many decision trees and thus are less susceptible to over-fitting. Different number of trees (5, 50, 100), maximum depth (5, 15, 30), and number of bins to be 10 were tested across all different random forest models.

Lastly, a support vector machine (SVM) model was also implemented, which attempts to divide the dataset into classes to find the best separating decision plane. Using 2 classes allowed provided good reason to use SVM as a classifier. Different settings including number of iterations (100, 200) and including an intercept are tested on the model.

3 Results

3.1 Models Constructed with Tuning Hyperparamters

Classification models were built with logistic regression, random forest, and SVM with default setting as the base.

Accuracy and precision were the two values that were most significant in the project. Accuracy counts for correct decisions in both classes, which shows the overall performance of the model. Precision shows how much true positive results are in all positive predictions. This was important in the problem because the prediction of a hotel booking can be based on the positive predictions.

Table 3.1 shows the model metrics for the different models that were run by the group. With just these three base models, it appeared that logistic regression had the highest and best accuracy among the three models. This made intuitive sense as logistic regression is a powerful method for binary classification.

Table 3.1: Model metrics for the original classification models.

	Accuracy	FPR	FNR	Precision	Recall	F_1	AUC
Logistic Regression	.813	.193	.171	.624	.829	.712	.818
Random Forest	.806	.196	.195	.618	.805	.699	.804
Support Vector Machine	.548	.164	.555	.883	.445	.592	.640

3.2 Model Selection

For logistic regression, different versions of the model were further tested by tuning the different hyperparameters (see Table 3.2), such as the iteration size and including the intercept. However, the accuracy for the model remained similar across testing. The model changed in accuracy when the number of iterations was increased from 10 to 100, but when the number was increased from 100 to 200, the result did not change much. Adding the intercept does not increase the performance significantly.

Therefore, the logistic regression model with iteration=100 and with an intercept resulted in the highest accuracy score, at around 81.3%.

Table 3.2: Model evaluations for Logistic Regression.

	Accuracy	FPR	FNR	Precision	Recall	F_1	AUC
Base model: no intercept, iteration=100	.8128	.193	.171	.624	.829	.712	.818
Iteration=100, with intercept	.8131	.194	.169	.622	.831	.712	.819
Iteration=10	.7998	.195	.213	.630	.787	.700	.796
Iteration=10, with intercpt	.7993	.196	.213	.628	.788	.699	.796
Iteration=200, no intercpt	.8129	.194	.170	.623	.830	.712	.818
Iteration=200, with intercpt	.8128	.194	.170	.623	.830	.712	.818

Table 3.3 shows the different model evaluation metrics for the random forest models. The base random forest model with no intercept gave poor results of accuracy at 55.8%. Increasing the number of trees increased the accuracy slightly, but will take a great amount of space in calculation, which was not possible in this project.

Otherwise, increasing the maximum search depth increased both accuracy and precision significantly. Among all the tested cases, the best random forest model utilized a depth of 30 to give 61% accuracy and a precision of 80.1%.

Table 3.3: Model evaluations for Random Forest.

	Accuracy	FPR	FNR	Precision	Recall	F_1	AUC
Base model: No of Trees=5, max depth=5	.558	.303	.160	.613	.839	.708	.768
Max. Depth=30	.611	.192	.169	.801	.830	.815	.819
No. of Trees=50, Max. Depth=15	.609	.219	.137	.751	.862	.803	.822
No. of Trees=100	.574	.296	.103	.608	.897	.725	.801

The third model chosen was SVM with stochastic gradient decent. Table 3.4 shows all the tested cases for SVM, and it was discovered that larger number of iterations slightly increased the accuracy, but it was not significant, and it was acceptable to choose a lower iteration number in consideration of computation capacity. The regularizer was not affecting the results significantly, so it should not be considered in the final design.

With the intercept, the model provided a higher accuracy score, but lowered precision to around 3%. This meant a great amount of false positive results were generated, which was not expected in our project, since false positive results will cause risk of overbooking for the hotel. Therefore, a model without intercept should be chosen for SVM.

Table 3.4: Model evaluations for Support Vector Machine.

	Accuracy	FPR	FNR	Precision	Recall	F_1	AUC
Base model: iteration=100, no intercept	.548	.164	.555	.883	.445	.592	.640
Iteration=100, with intercept	.638	.364	.257	.034	.744	.066	.690
Iteration=200, no intercept	.554	.168	.552	.875	.448	.593	.640
Iteration=200, with intercept	.633	.368	.274	.015	.726	.029	.679
Iteration=200, with intercept regType=’l1’	.634	.367	.236	.019	.764	.038	.699
Iteration=100, with intercept regType=’l1’	.638	.364	.255	.034	.745	.066	.690

Table 3.5: Best model from each category.

	Accuracy	FPR	FNR	Precision	Recall	F_1	AUC
Logistic Regression: with intercept, iteration=100	.813	.194	.169	.622	.831	.712	.819
Random Forest: Max. Depth=30	.611	.192	.169	.801	.830	.815	.819
SVM: no intercept, iteration=100	.548	.164	.555	.883	.445	.592	.640

From table 3.5, we can found logistic regression to be the best performing approach over the random forest and SVM alternatives with regard to accuracy, recall, and area under the ROC curve (AUC). Although the SVM model with 100 iterations obtained the best precision score of 88.3%, which is important in minimizing overbooking potential, the poor model accuracy made initial predictions that were poor indicators of whether a patron had actually cancelled the booking reservation. Therefore, the logistic regression model with intercept term and 100 iterations was our champion model with an accuracy of 81.3%.

4 Conclusions and Future Directions

4.1 Conclusion

In conclusion, a 2-class classification model can be used to make predictions on the potential cancellation of hotel bookings. From the accuracy and precision perspective, we chose the model that worked best for the problem. With tuning of model parameters, logistic regression was found to be the best model, because it has a good accuracy, decent precision, and a fast calculation speed. To interpret the results, the chosen logistic regression model resulted in accurate predictions with less probability of overbooking a reserved room.

4.2 Limitations and Future Directions

There are some limitations that need to be addressed that relate to the nature of the dataset and the tools that were used. First, the dataset had a limited size of variables and there could be latent variables that did not exist in the dataset that affected the predictions. With more variables, the predictions may be more reliable. Second, the server had limited memory for training for many of the current models, so the choice of models and training settings were constrained due to time and computational limits. Finally, Spark does not have visualization tools, which made creating figures difficult and required hard coding numeric data to fit `Matplotlib` functions.

If given more time for future evaluation, the group can improve the results in several directions. First, the weights of the variables should be considered when training the model. Considering the importance of each variable can fit the models better with the data. Second, more tuning parameters can be used to optimize the model. Third, the results can be shown with more visualizations to convey information better. Lastly, implementing k-fold cross validation when training the models could give us more metrics and can be used to fine tune our parameters.