DATA 1030 Final Project Report: Prediction Analysis of Hotel Reservation Cancelation Based on Extracted Features

Siyuan Li
Brown University
Siyuan_li2@brown.edu
GitHub-link: https://github.com/siyuanli1202/DATA_1030_FINAL_PROJECT

Abstract: Within the report, we will perform a detailed data analysis for one real multivariate dataset related to hotel bookings. The data explored contain both continuous and categorical variables. The report will extend the exploration of various classification and regression methodologies to predict the probability rate of a cancelation. The analysis also intends to enhance the prediction accuracy by reducing the data dimension with the extracted features that have the most contribution to represent the fitted model.

I. Introduction

The project is concentrated on the analysis of "*hotel_bookings.csv*" data. The resource is originally from the article Hotel Booking Demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, and cleaned by Thomas Mock and Antoine Bichat for research purpose. The data collected 119390 pieces of users' hotel booking information (data points), and 32 related features are recorded for each piece of information. In our analysis, we hope to use machine learning techniques to predict whether users would cancel their hotel reservations. Therefore, the target variable is a column in our dataset named "*is_canceled*". The aim for initialize the research project to solve the issue is to provide insights for hotel managers or administrators to better arrange their business operations. We would understand more clearly about the factors which affect users to retrieve their bookings. Currently, many professional projects are working on the same topic as ours, which is to predict the likelihood of a booking to be canceled. Their works shed insights on utilizing various classification models such as Random Forest and XGBoost to improve the prediction accuracy. We summarized their attributes in the following table.

Categorical Features ***(15 counts)***	**Continuous Features** ***(17 counts)***
1. 'hotel' 2. 'is_canceled' 3. 'arrival_date_year' 4. 'arrival_date_month' 5. 'meal' 6. 'market_segment' 7. 'distribution_channel'	1. 'lead_time' 2. 'arrival_date_week_number' 3. 'arrival_date_day_of_month' 4. 'stays_in_weekend_nights' 5. 'stays_in_week_nights' 6. 'adults' 7. 'children'

8. 'is_repeated_guest'	8. 'babies'
9. 'reserved_room_type'	9. 'previous_cancellations'
10. 'assigned_room_type'	10. 'previous_bookings_not_canceled'
11. 'deposit_type'	11. 'booking_changes'
12. 'customer_type'	12. 'agent'
13. 'required_car_parking_spaces'	13. 'company'
14. 'total_of_special_requests'	14. 'days_in_waiting_list'
15. 'reservation_status'	15. 'adr'
	16. 'required_car_parking_spaces'
	17. 'total_of_special_requests'

Table 1

II. EXPLORATORY DATA ANALYSIS

As we move to the next-step exploration of the data analysis, the key part we need to perform is EDA. To obtain a general view of the dataset, we apply the Python in-built function ‘*.describe or .value_counts’* to summarize the 32 features of booking information. We also create histograms and bar plots for each feature based on their attributes respectively. This statistical result is presented in the notebook. For simplification, we intended to demonstrate three interesting plots, which help us to understand the correlation of target label and rest features with a better angle. graphs are shown as below.

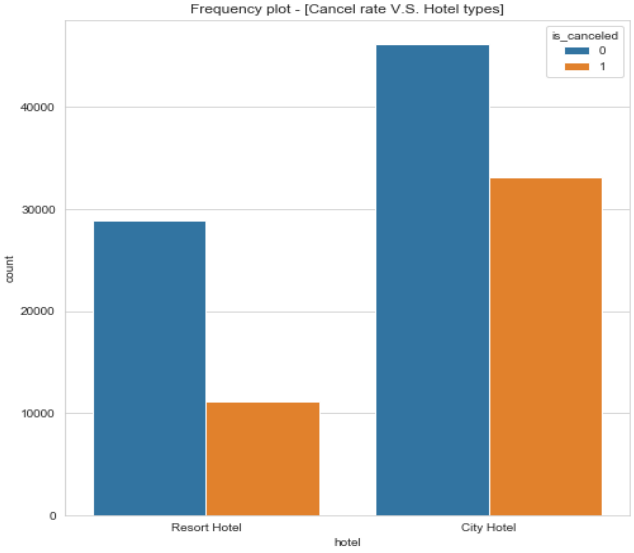

Figure 1 - Frequency plot - [Cancel rate V.S. Hotel types]

The first plot is a bar plot, which describes the relationship between booking cancelation frequency and hotel types. We can see that there is quite a visible difference between the two hotel types: city hotel and resort hotel. The cancelation appeared more frequently at city hotels, and the reason might be caused by customer’s motivation of booking. People usually book a resort hotel for spending their vacations, while more for business purposes when choosing city hotels. The difference in indentation might bring the result.

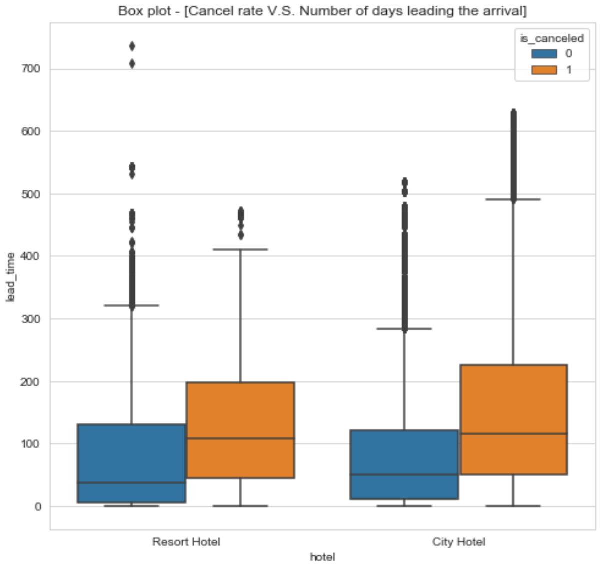

Figure 2 Box plot - [Cancel rate V.S. Number of days leading the arrival]

The second plot is a box plot, which characterized the relationship between booking cancelation frequency and leading time before the arrival data. More specifically, we plotted the correlation according to hotel types. In general, we can see that there is not an obvious difference between city hotels and resort hotels. However, the plot shows us that there is a trend appearing in both two categories. As customers' leading time increased, they are less likely to cancel their reservations. The trend might be due to the fact that if people plan their trips or booking very early, they would be more determined to finish the trip and therefore take the bookings. In this way, we understand that leading time plays an important role in deciding whether a reservation would be canceled or preserved.

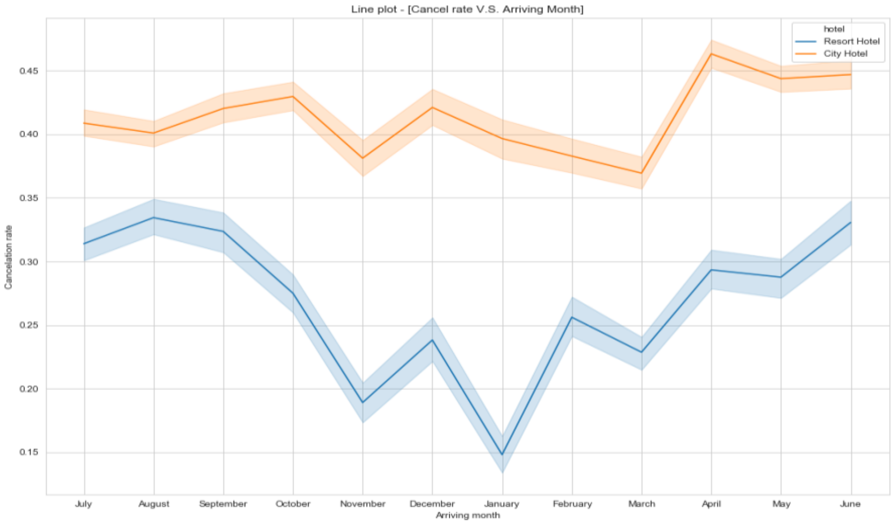

Figure 3 - "Line plot - [Cancel rate V.S. Arriving Month]"

The third plot is a line plot, which depicted the fluctuation of between booking cancelation frequency and month in a year. The plot is more designed to understand the time factor’s influence on our prediction problem. With the line plot, we can discover that customers are more likely to cancel their reservation in the month like June and July, less likely in January, November and December. Accordingly, we can consider that time serves as a key factor of prediction problem.

III. METHODS

As mentioned before, our prediction problem is to predict the likelihood of hotel bookings cancelation. We first clean the dataset by removing the "nan" values, which reduce a small portion of data points by 13.81%. Then we explore the dataset dimension, and find that the data frame contains about 118898 pieces of information. As the data set can be perceived as medium-sized, we will apply the traditional splitting methodology as proportion divide into 60% training, 20% validation, 20% test groups. Reflecting on the data collection source, we understand that our data is in i.i.d form. Meanwhile, there are no group structures or time series related data existing in the hotel bookings information. Given those conditions and that the data is balanced, we would directly split the dataset and apply encoders on each feature. In our pipeline construction, the splitting process is repeated for five times for balancing the randomness.

Now, we lead our discussion to the type of encoders utilized. Our label variable is “is_canceled”, which is binary categorical variable. As the label has already been simplified into two categories, there is not necessary for now to apply “LabelEncoder” on the target. For all categorical features, we defined in the above table, we utilized “OneHotEncoder” as those features do not include ranking information. For continuous variables, we need to divide into two situations: (1) We apply “MinMaxEncoder” for two features named 'arrival_date_week_number', and 'arrival_date_day_of_month', because they are bounded in a fixed interval. (2) For the rest of continuous features, we directly use “StandardScaler”. The final prepossessed training feature dataset is increasing from 29 to 1143 columns, where we have transformed 15 categorical features.

After prepossessing progress, we move to the implementation of different machine learning models. Note that our problem is belong to the classification analysis, we would first like to compute the baseline accuracy score for the prediction. For evaluation metric, we choose accuracy score for our method. The reason is that our data is quite balanced, with roughly 40:60 distribution of two predicted classes. Meanwhile, we believe the correct prediction for both two classes means equally important to us. Thus, the accuracy metric is selected.

By continuous exploration, the baseline accuracy rate, the portion of larger class between the two, is equal to 60.96%. Then, considering our training dataset’ dimension is quite large, we choose four methodologies and fit them with a random training dataset to decide which model would be further researched. We summarize the first-round prediction result as below.

Machine Learning algorithms	**Logistic Regression**	**Random Forest**	**KNN Classification**	**XGBoost**
Accuracy score	0.8516	0.8136	0.8514	0.9230
Time Consumption	Fast	Very Fast	Very Slow	Slow

Table 2

Since our data is in high dimensional form, the selection of machine learning models will be essentially according to both prediction result and running efficiency. In this analysis, we choose three algorithms for hyperparameter tuning, which are Logistic Regression, Random Forest and XGboost. The following research result is discussed in the next part.

IV. Results

The analysis in this part will be divided into two parts. In the first part, we present the prediction results of three different selected models, and demonstrated the comparison with baseline accuracy. In the second part, we would discuss feature importance in a global perspective.

Part I Model Performance

(1) Logistic Regression

Scores\Solver	**'lbfgs'**	**'sag'**	**'saga'**
Training	0.8622	0.8282	0.8197
Validation	0.8544	0.8263	0.8180
Test	0.8537	0.8237	0.8161
Baseline accuracy	0.6096		

Table 3

From a rudimentary testament we leaded at the beginning of our analysis, logistic regression model can be considered as a good one, since it provided a descent accuracy rate and time consumption is short. We then explored the method a step further. The tuning parameter here is solver, which served an important role in predicting the binary labels. Although solver 'sag' and 'saga' are designed for big datasets, what we find interesting is that the default solver 'lbfgs' give us the best test score and prediction result. Compared to the baseline strategy, the improvement in prediction accuracy is 39.43%.

(2) Random Forest

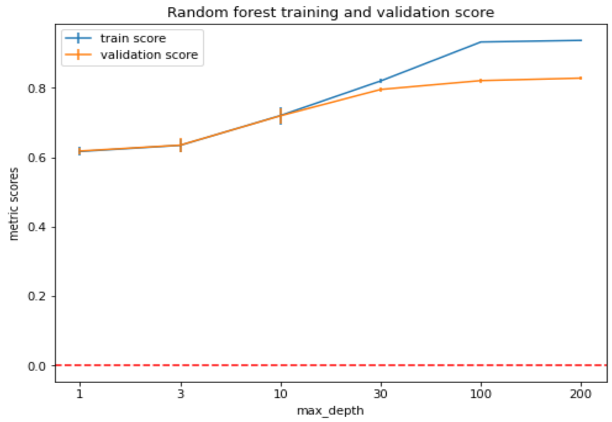

Figure 5

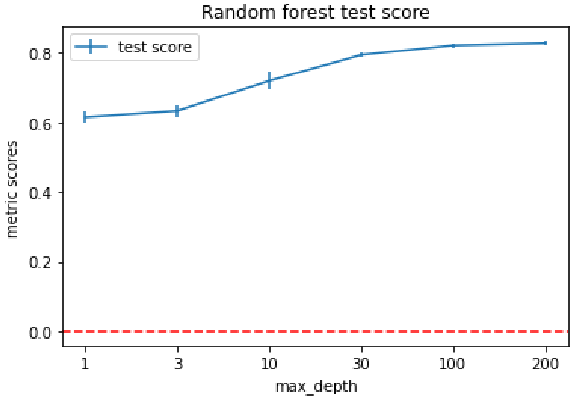

Figure 4

Random forest model is the fastest one among the starting four kinds. In its exploration, we are tuning the parameter ‘max_depth’ and use error plot to demonstrate the test scores’ fluctuation. We can clearly find that as the depth parameter increases, all three metric scores are improving. The fact indicating depth as an important factor, and we find that as depth approximate to 200. The improvement is slowing down. For this model, the final improved test accuracy score can reach 82.84%. Compared to the baseline strategy, the improvement in prediction accuracy is 35.89%.

(3) XGBoost

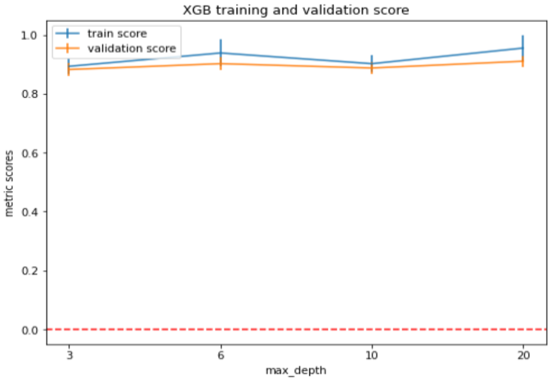

Figure 6

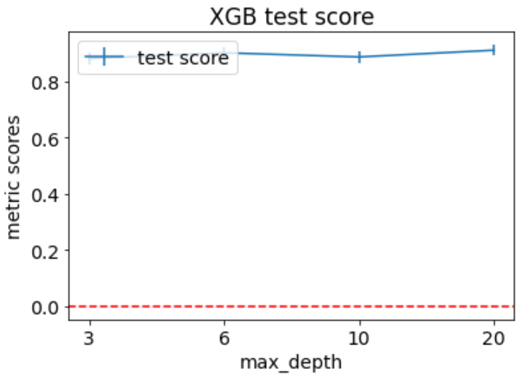

Figure 7

XGBoost model provided us a best accuracy score in the first model fit, but its running time is slower than both logistic regression and random forest methods. Given it is a very memory-costing strategy and data size is large, we choose only to tune the ‘max_depth’ parameter with four reasonable values. The result is presented with above plots, we can find several interesting discoveries here: (1) As the depth parameter increases, the train and validation score happen to have a decreasing interval from six to ten, but a new increase from ten to twenty. (2) The test is roughly without any huge change. The method gives us a very good prediction result, with the highest accuracy score equal to 91.02%. Compared to the baseline strategy, the improvement in prediction accuracy is 49.31%.

Part II Feature Importance

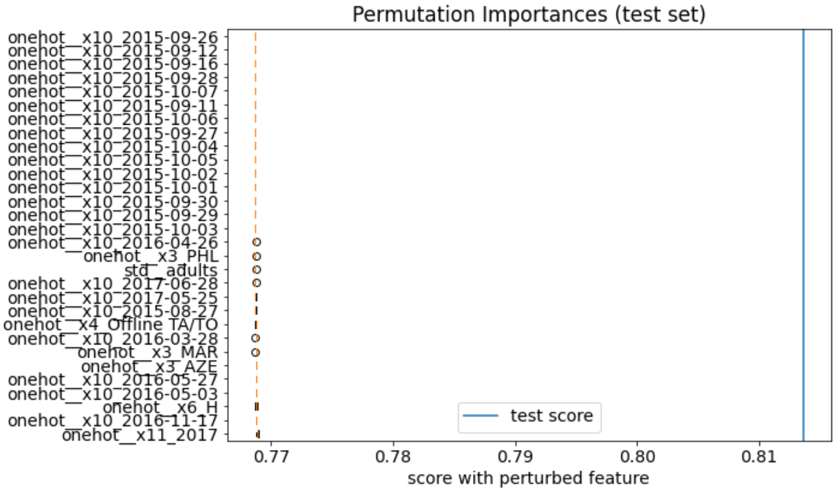

Figure 8

Since our data is consist of over 100,000 data points and 29 different column features, we choose not to look through their importance with a local scope, instead with a global permutation method. The model fit we selected is random forest, as it would help us to boost the efficiency of implementation the method. The plot (top20) can show us that the several features plays a decisive role in determining whether customers would cancel their reservation. We find the many columns in the plot is related to the feature ‘reservation_status_date’. This finding can be very interesting, because it indicates us certain date in a year can affect customers to manage their hotel reservations. Some other important features are also intriguing for us to illustrate, such as ‘adult’, ‘arrival_date_month’ and ‘country’. They show that the number of adults, arrival time and people’s nationality can be factors that affect the reservation.

V. OUTLOOKS

Reflecting on the research we carried for this problem, we could make many improvements in the next-step exploration. Starting with the random splitting process, we did not use the conventional cross-validation method to divide the training and validation datasets because of the heavy cost of computations. If we hope to increase the balance of randomness, we might to apply “GridSearchCV” strategy to implement the machine learning model fit. The prediction result of three models can be more conclusive with this adaption. Meanwhile, the tuning parameters can be enriched. Since for each machine learning pipeline, we only choose one important parameter for each model. In future, if we expanded the research of hyperparameter tuning, a better prediction model might appear. In the last, in the progress of permutation feature importance, we carried the random forest model because it has the fastest calculation speed. This limitation is produced by our hardware conditions. If we are allowed to test the feature importance with model that has the best prediction result, the returned result can be a meaningful discovery for us to understand.

References

[1] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

[2] Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.

[3] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science &amp; Engineering*, *9*(3), 90–95.

[4] McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

[5] Mock, T., &amp; Bichat, A. (2020, February 11th). [The data is originally from the article Hotel Booking Demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.]. Published raw data.

[6] Brownlee, J. (2020, August 27). Tune Hyperparameters for Classification Machine Learning Algorithms. Retrieved November 30, 2020, from https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/

[7] Czakon, J. (2020, January 16). The ultimate guide to binary classification metrics. Retrieved November 30, 2020, from https://towardsdatascience.com/the-ultimate-guide-to-binary-classification-metrics-c25c3627dd0a

[8] Hale, J. (2020, April 07). Don't Sweat the Solver Stuff. Retrieved November 30, 2020, from https://towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451

[9] Haupt, J. (2020, July 20). Https://johaupt.github.io/scikit-learn/tutorial/python/data%20processing/ml%20pipeline/model%20interpretation/columnTransformer_feature_names.html.