Hotel Booking Analysis: Cancellation Prediction

Rebekah Rest
Brown University
Github: https://github.com/rebekahrest/data1030-final-project

Introduction

Purpose

Hotels suffer from high cancellation rates, which negatively impacts their efficiency when resources and personnel are underutilized. Hotels lose money from customer cancellations, especially when they occur at the last minute. Hotel owners can benefit from being able to predict individual cancellations, which would help them optimize their operational strategies such as more accurate staffing and food requirements. Hotels might also increase profits by overbooking rooms according to the predicted cancellations, similar to what the airline industry already does.

Hotel Booking Demand Dataset

The hotel booking demand dataset contains 119,390 data points with hotel booking information between the dates 7/1/15 to 8/31/17. The data was collected from two hotels in Portugal, a resort hotel in Algarve and a city hotel in Lisbon. The resort hotel had 40,060 data points, and the city hotel had 79,330 data points. [1] The data was collected via TSQL queries executed directly in the hotels' property management system SQL databases. There are 31 variables describing the data points, and all data elements describing hotel or customer identification were deleted. [2]

Target Variable and Features

The hotel booking demand dataset is a classification dataset. The target variable is 'is_canceled,' with '0' indicating not canceled, and '1' indicating a cancellation. There are 119,390 rows and 31 columns. The 30 features are:

lead time, arrival date year, arrival date month, arrival date week number, arrival date day of month, stays in weekend nights, stays in week nights, adults, children, babies, meal, country, market segment, distribution channel, is repeated guest, previous cancellations, previous bookings not canceled, reserved room type, assigned room type, booking changes, deposit type, agent, company, days in waiting list, customer type, average daily rate, required car parking spaces, total of special requests, reservation status, and reservation status date.

Previous Work

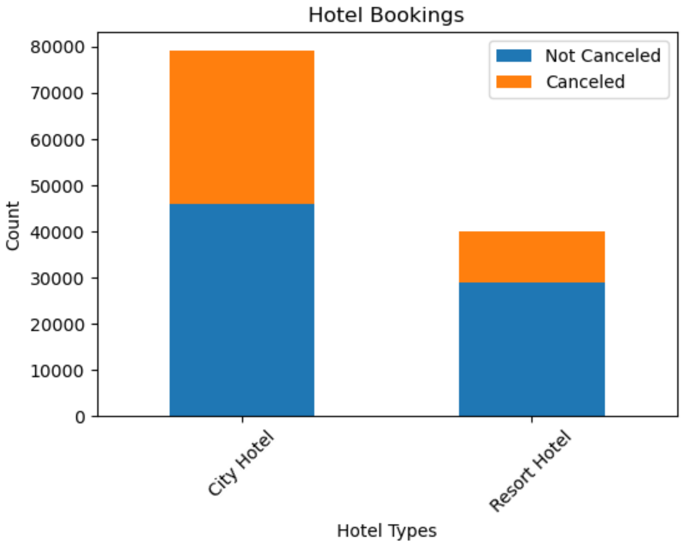
The researchers who constructed the hotel booking demand dataset used a dataset of 73,131 points collected from 4 hotels in Portugal with 37 variables to build models for predicting booking cancellations. Using a boosted tree model and a decision forest, they were able to achieve accuracies between 91%-98% for all 4 hotels. [3]

Exploratory Data Analysis

I used .describe to analyze the distributions of the features and to check for null values. According to the paper, the dataset has no missing values and null should be treated as ‘N/A.’ I made bar plots, histograms, violin plots, and box plots for analysis of the features.

Hotel Types and Cancellations

The target variable is ‘is_canceled,’ with ‘0’ indicating the booking was not canceled and ‘1’ indicating a cancellation. There was an overall cancellation rate of 37%, with 28% of resort hotel bookings canceled and 42% of city hotel bookings canceled.



<Bar Plot of Cancellations by Month>

Bookings:

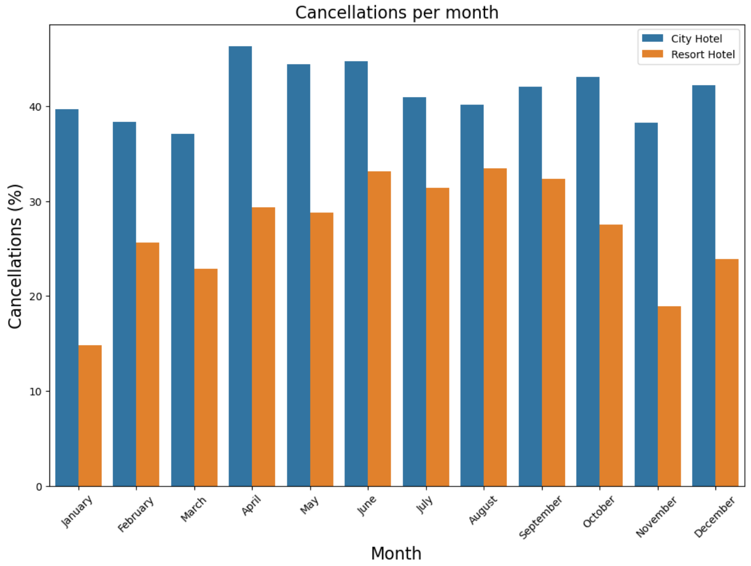
- City hotel: 79,163
- Resort hotel: 40,047

Cancellations:

- Total bookings canceled: 44,199 (37%)
- Resort bookings canceled: 11,120 (28%)
- City bookings canceled: 33,079 (42%)

Cancellations By Month

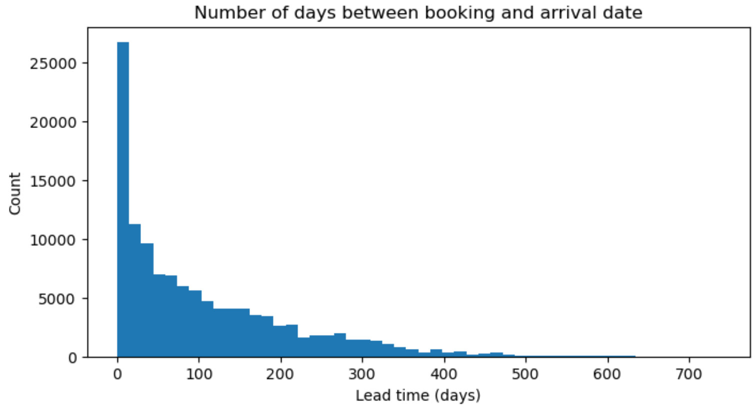
The bar plot of cancellations by month shows that the city hotel cancellations remained about the same for each month at 40%, while the resort hotel cancellations were highest in the summer, likely with vacationers, and lowest in the winter. This is important for predictions later, with the month of the booking important for predicting cancellations for the resort hotel.

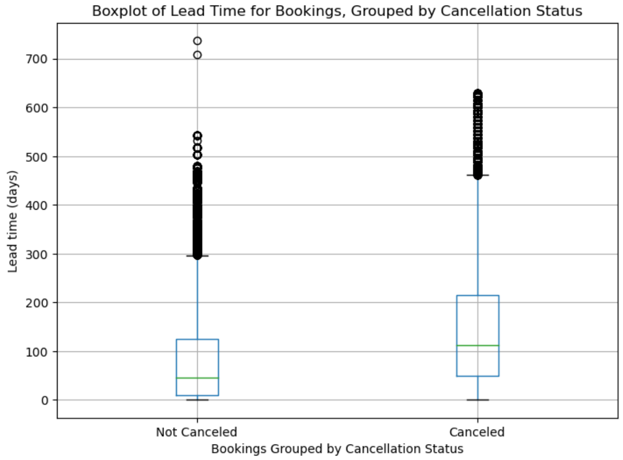


<Bar Plot of Cancellations per Month>

Lead Time

The lead time is the number of days between the booking and the arrival date of the reservation. The distribution of lead time is right skewed, with many bookings made with short lead time, with a mean around 100 days. From the box plot, the canceled bookings have a slightly higher average lead time than bookings that weren't canceled. This makes sense, as it is expected bookings made earlier have more uncertainty and are more likely to be canceled.



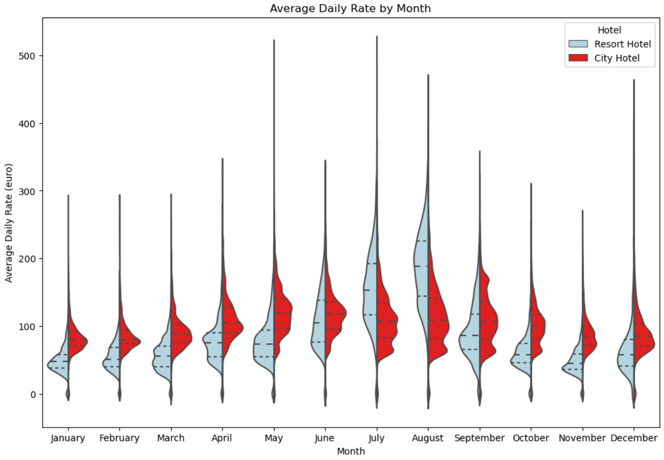


<Histogram of Lead Times of Bookings>

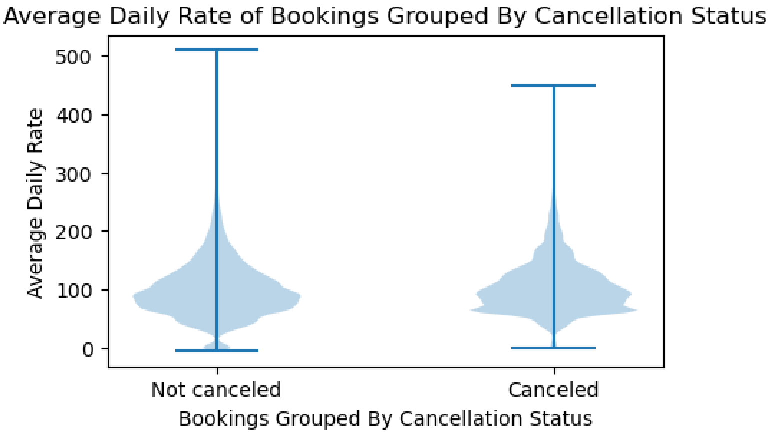
<Box Plot of Lead Times by Cancellation Status>

Average Daily Rate

The average daily rate plots show that the resort hotel was cheaper than the city hotel except in July and August, when it was significantly more expensive. Overall, hotels that were canceled had a higher mean average daily rate at € 105 vs. € 100.



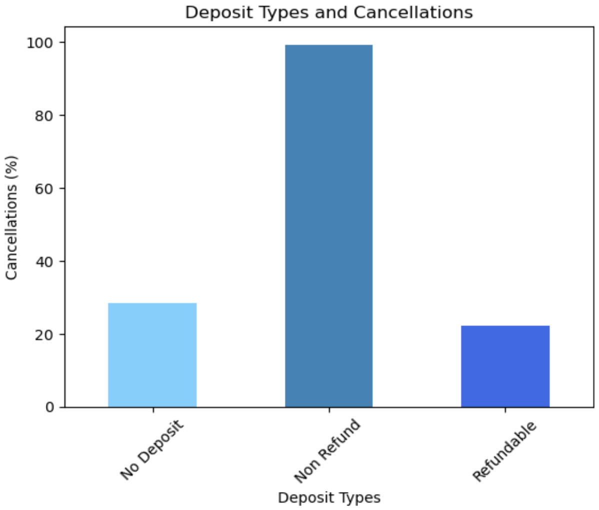
<Violin Plot of Average Daily Rate of Bookings by Month>



<Violin Plot of Average Daily Rate of Bookings Grouped By Cancellation Status>

Deposit Type

The deposit type bar plot showed surprising findings. Most bookings were made with no deposit, with a small number of non refundable and refundable bookings. However, almost all non refundable bookings were canceled, with a rate of 99%, which is quite strange. The author of the dataset suggested that these bookings were used to support visas to enter the country, since a hotel booking is required for entry. They likely used invalid credit card credentials, which the hotel later in the verification process identified as fake and canceled. [4]



<Cancellation Rate of Bookings Grouped By Deposit Type>

Methods

Splitting

The data was first split using sklearn’s train_test_split 80:20 into X_other and X_test, then KFold with 4 folds is applied on X_other. This results in a 60:20:20 split between the train, validation, and test sets. Using KFold Split with 4 folds is effective but still manageable with a large dataset. It also reduces the variance and bias of performance estimates and prevents overfitting. A standard KFold split is used because the dataset is not overly imbalanced to necessitate a Stratified Split, and a group-based split would not make sense for the purpose of predictions on individual bookings.

Preprocessing

To prevent data leakage, I dropped two features, the reservation status, which gave the same information as the target variable, is_canceled, and booking_changes, as this could change over time.

For scaling and preprocessing the features, I used a Column Transformer with a OneHotEncoder and OrdinalEncoder for the categorical features, and a StandardScaler and MinMaxScaler for the numerical features.

> **OneHot Features:** "hotel", "meal", "market_segment", "company", "assigned_room_type", "distribution_channel", "reserved_room_type", "deposit_type", "customer_type", "is_repeated_guest", "country"
>
> **Ordinal Features:** "arrival_date_month"
>
> **Standard Features:** "lead_time", "arrival_date_year", "stays_in_weekend_nights", "stays_in_week_nights", "adults", "children", "babies", "previous_cancellations", "previous_bookings_not_canceled", "days_in_waiting_list", "required_car_parking_spaces", "total_of_special_requests", "adr", "agent"
>
> **MinMax Features:** "arrival_date_week_number", "arrival_date_day_of_month"

The transformation changed the dimensions of the training set from (76295, 28) to (76295, 569).

ML Pipeline

I used sklearn's Pipeline with two steps: the preprocessor and model. I trained four machine learning algorithms: Logistic Regression, K Nearest Neighbors, XGBoost, and Random Forest. For each model, the following parameters were tuned using GridCV:

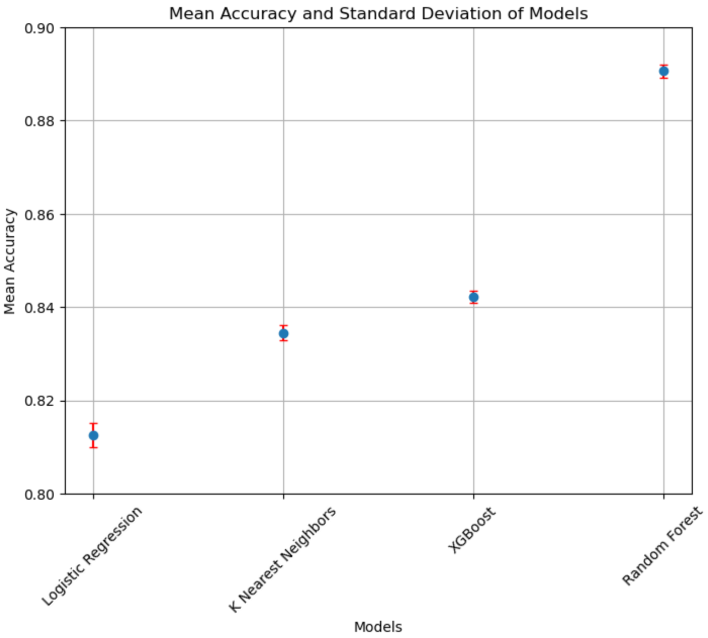
Model	Hyperparameters:
Logistic Regression	C : [0.001, 0.01, 0.1, 1, 10, 100]
K-Nearest Neighbors	n_neighbors : [3, 10, 30, 100]
XGBoost	max_depth : [1, 3, 10, 30, 100]
Random Forest	max_depth : [10,30,100,300] max_features: [0.5,0.75,1.0] min_samples_split : [2,4,8,16,32]

The metric used to evaluate the models' performance was accuracy. The dataset is relatively balanced, with 75,011 or 63% of points not canceled, and 44,199 or 37% of points canceled. Using f1 score is not necessary, and accuracy has the advantage of being easy to understand. Uncertainties from splitting and non-deterministic methods were addressed by using 5 different random states for splitting when training the models and using KFold cross validation with 4 folds.

Results

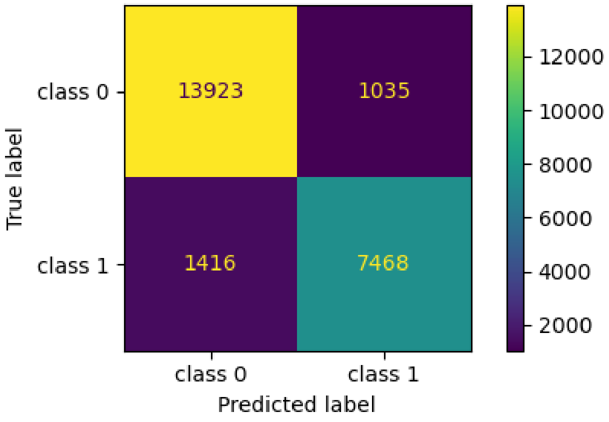
Based on the mean accuracy of the models' performance on the test set, the Random Forest model performed the best, while XGBoost, KNN, and Logistic Regression performed slightly worse. The Random Forest model showed a 0.8906 mean test accuracy. All models outperformed the baseline accuracy of 0.63, the proportion of the majority class '0.'

Model	Mean test score (accuracy)	Standard Deviation	Z-Score: (Mean test score - baseline) / std_dev
Baseline (Majority Class=0)	0.63	0.0014	___
Logistic Regression	0.8125	0.0026	130.88
K Nearest Neighbors	0.8345	0.0016	146.62
XGBoost	0.8422	0.0014	152.13
Random Forest	0.8906	0.0014	186.83

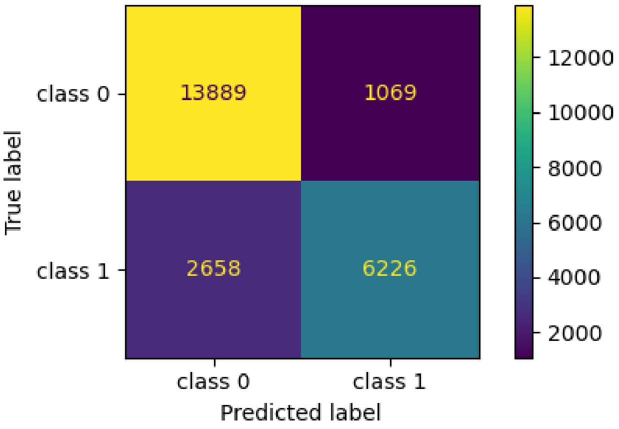


<Plot of Mean Accuracy and Standard Deviation of Models>

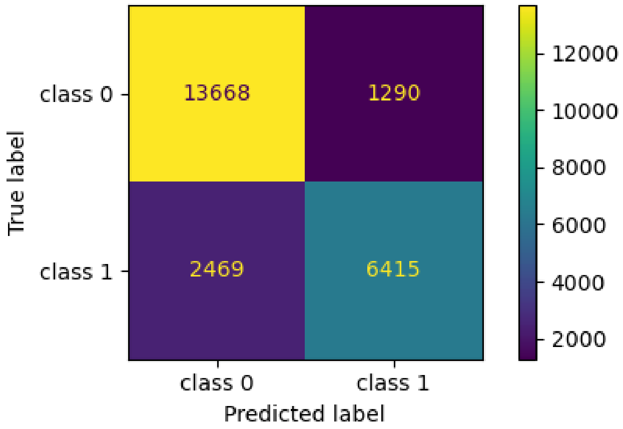
From the confusion matrices of the models, the Random forest performed the best with the highest accuracy and the smallest false positive and false negative rates. For a hotel booking prediction model, minimizing the false positive rate is important if the hotel wants to act on bookings that will be canceled. The lower the false positive rate is, the less the hotel will spend on unnecessary preparations for a booking that doesn't get canceled.



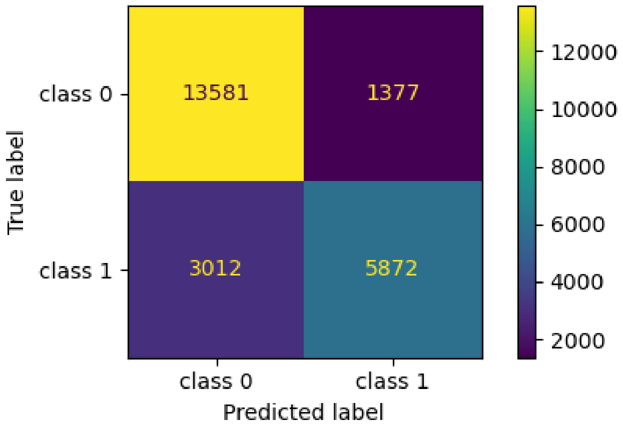
<Confusion Matrix for Random Forest>



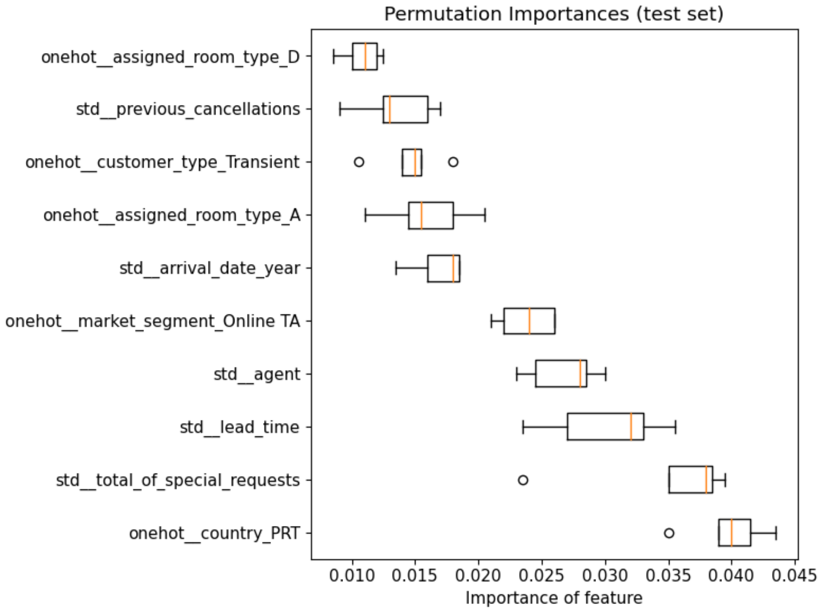
<Confusion Matrix for XGBoost>


True label
class 0
class 1
13668
1290
2469
6415
class 0
class 1
Predicted label
12000
10000
8000
6000
4000
2000

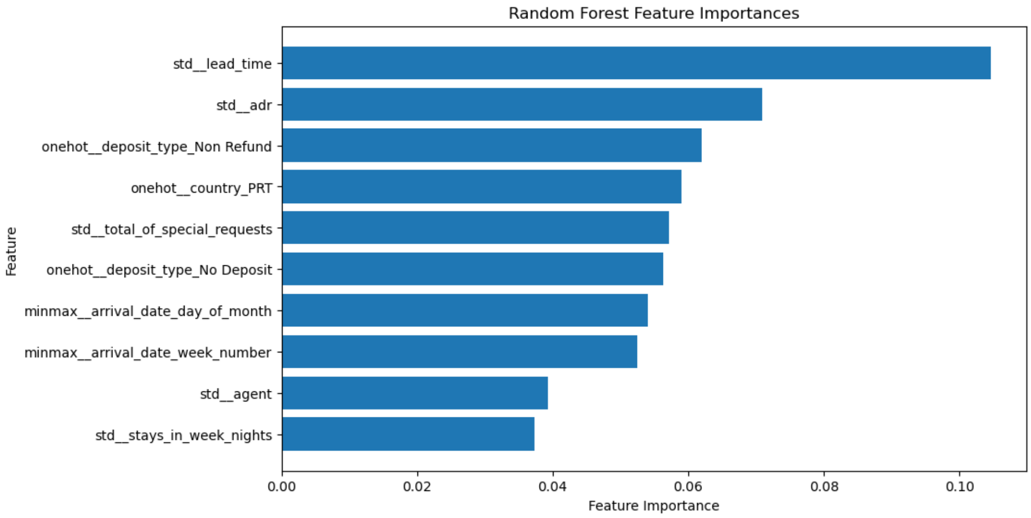
<Confusion Matrix for KNN>


True label
class 0
class 1
13581
1377
3012
5872
class 0
class 1
Predicted label
12000
10000
8000
6000
4000
2000

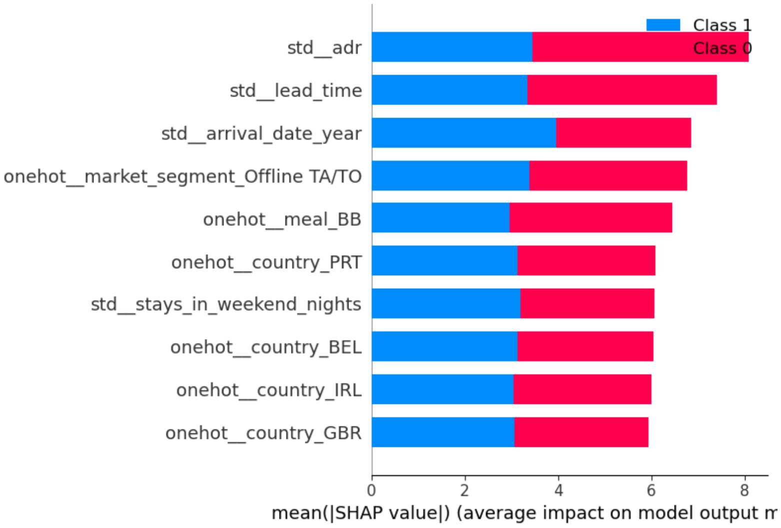
<Confusion Matrix for Logistic Regression>


Permutation Importances (test set)
onehot__assigned_room_type_D
std__previous_cancellations
onehot__customer_type_Transient
onehot__assigned_room_type_A
std__arrival_date_year
onehot__market_segment_Online TA
std__agent
std__lead_time
std__total_of_special_requests
onehot__country_PRT
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
Importance of feature

<Top Ten Features by Permutation Importance>

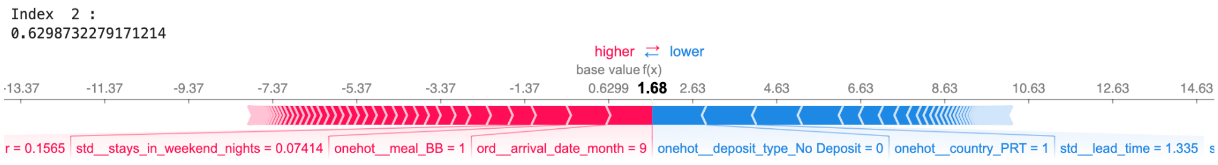


<Top Ten Features by Impurity-Based Importance>



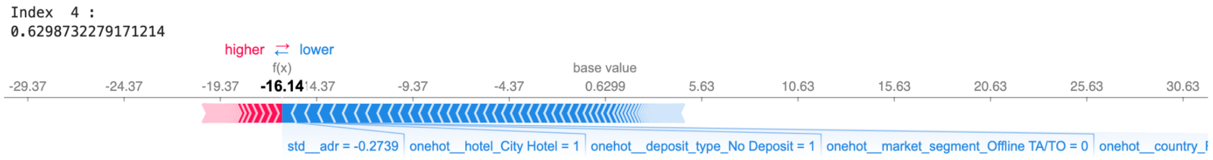
<Top Ten Features By Shap Value>

Permutation importance, impurity-based importance, and Shap value were used to determine the importance of feature contributions to the prediction of the Random Forest model. According to Permutation importance, the top three features were whether a guest's country of origin was Portugal, the total number of special requests, and the lead time. According to impurity-based importance, lead time, the average daily rate, deposit type of non refundable were most important. And from Shap, the average daily rate, lead time, and arrival date year were the top three. Most features in the top ten contributed more towards class 0, while arrival date year was more significant for class 1. Overall, the three metrics ranked lead time and the average daily rate highly in feature importance, which corresponds with findings from the exploratory data analysis.



```
Index  2 :
0.6298732279171214
```

<Shap Local Feature Importance for Point at Index 2>



```
Index  4 :
0.6298732279171214
```

<Sha Local Feature Importance for Point at Index 4>

The figures show feature explanations for index 2, belonging to class 1, and index 4, belonging to class 0. For index 2, the deposit type and arrival date month contributed highly, with other globally important features including country of origin Portugal and lead time playing a slightly less significant role. For index 4, the average daily rate and type of hotel were most important, with deposit type, market segment offline TA/TO, and country of origin Portugal contributing as well.

Outlook

The Random Forest model had the best performance with a max depth of 30, min samples split of 2, and max features of 0.5. Additional parameters that could be trained include n_estimators, min_samples_leaf, and criterion to improve accuracy and make the model more robust. The dataset was large, which made hyperparameter tuning difficult. The other models had only one hyperparameter tuned due to long run times, so with more time I would tune additional hyperparameters to improve the accuracy. Additional models could also be tested. I trained a preliminary baseline SVC model with a mean test accuracy of 0.8488, but the prohibitively long run time made it impossible to tune hyperparameters, so I did not include the model in the final set of models. Further, for more model explainability LIME can be used to explore feature importances.

References

[1] Mostipak, J. (2020, February 13). *Hotel Booking Demand*. Kaggle. https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand/data

[2] Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel Booking Demand Datasets. *Data in Brief*, *22*, 41–49. https://doi.org/10.1016/j.dib.2018.11.126

[3] Antonio, N., Almeida, A. de, & Nunes, L. (2017). Predicting Hotel Booking Cancellations to Decrease Uncertainty and Increase Revenue. *Tourism & Management Studies*, *13*(2), 25–39. https://doi.org/10.18089/tms.2017.13203

[4] Antonio, N., de Almeida, A., & Nunes, L. (2019a). Big Data in Hotel Revenue Management: Exploring cancellation drivers to gain insights into booking cancellation behavior. *Cornell Hospitality Quarterly*, *60*(4), 298–319. https://doi.org/10.1177/1938965519851466