MSiA 420: Final Project - Group 3

Alejandra Lelo de Larrea Ibarra Kiran Jyothi Sheena Lixuan Chen
Wencheng Zhang

Executive Summary

This report details the findings of a class project for MSiA 420, which aimed to predict the cancellation status of hotel bookings using machine learning classification models. The project utilized a Portuguese hotel data set available on Kaggle and tested nine models, including Naive Bayes, Logistic Regression, KNN, Neural Network, Decision Tree, GAM, SVM, XGBoost, and Random Forest. We evaluated the models based on their misclassification rate, F-1 score, ROC AUC, and training speed. The results showed that the Random Forest model was the best-performing model for predicting the cancellation status of hotel bookings, with a test misclassification rate as low as 11.7%. The Neural Network and XGBoost models also performed well. Although Naive Bayes was the fastest model, its accuracy was one of the worst among the nine models tested. The Random Forest model was both accurate and relatively fast, while XGBoost was even faster and still relatively accurate. These findings have important implications for hotel management and policymakers looking to develop effective strategies for managing cancellations and optimizing revenue streams.

Contents

1 Introduction

Smith Travel Research estimates there are 17.5 million guestrooms in 187,000 hotels worldwide. In 2020, global hotel revenue was $198.6 billion dollars, and the hotel and tourism industry accounts for approximately 10% of worldwide GDP (Hollander Accessed March 3rd 2023). The hotel industry experiences approximately 24% of cancellations on reservations, and this rate increases up to 38% for online bookings (Loeb Accessed March 3rd 2023). With these figures, and especially after the Covid-19 Pandemic, the study of hotel booking cancellations has become more relevant.

Accurately forecasting hotel booking cancellations and understanding the factors that influence such behavior is relevant for the following three reasons. First, from an economic perspective, each unoccupied room results in an monetary loss for the hotel. If accurately and timely forecasted, the hotel could still allocate the room to a different customer and avoid the loss of revenue. Additionally, an accurate prediction model could help the hotel to understand their net demand. Second, from the operational side of the business, managing cancellations is costly both in time and resources. For example, during peak seasons the hotel might need to allocate additional resources to manage cancellations and to try to reallocate the rooms. An accurate forecast could help to reduce operational costs and improve the hotel's efficiency. Third, from a marketing perspective, forecasting booking cancellations opens the possibility to implement pricing strategies, such as offering discounts to retain the customer.

Hotel booking cancellations and revenue management has been widely studied in literature. (Antonio, Almeida, and Nunes 2019) use data from eight hotels combined with additional sources (such as weather or holidays) to develop booking cancellation prediction models to help hoteliers understand their net demand and improve their revenue management. (Chen, Schwartz, and Vargas 2011) use a multi-nomial logit regression model to analyze the impact of cancellation fees and deadlines on hotel bookings and find that the former affects customer's behavior but the latter doesn't. (Falk and Vieru 2018) use data from 9 hotels to estimate a Probit model with cluster adjusted standard errors at the hotel level to find the determinants of cancellation probability, among which booking lead time and country of residence are the most important. (Sanchez-Medina, Eleazar, et al. 2020) apply ML algorithms to forecast hotel booking cancellations using genetic algorithms to configure the structural parameters of the artificial neural network used.

In this regard, we contribute to the study of hotel booking cancellations by analyzing the *Hotel booking demand* data set available in Kaggle[1] with the aim of forecasting hotel booking cancellations through the lens of Machine Learning (ML) models. This data set contains 119,390 booking records of a resort hotel and a city hotel from a chain in Portugal, which includes 32 predictors such as the date of the booking, length of stay, number of children and adults, type of meal plan, number of special requests, among others. To forecast cancellations, we first perform an exploratory data analysis to understand the data set and the relationships between the predictors and the response. Then we clean the data, select the main features and apply some transformations, as well as create new features. Then we fit several ML classification models to the final data set. For this process, we randomly split the data in training (95,511 observations) and test (23,877 observations), but keeping the same training/testing split across different models for fair comparisons. Specifically, we estimate a Naive Bayes classifier (benchmark), Logistic Regression, Neural Network, Tree, k-Nearest Neighboors, GAM, XGBoost, Random Forest and Support Vector Machines. For each of these models, we use 10-fold cross validation (CV) for hyperparameter tuning. Note that each model is run independently in this step. After finding the optimal hyperparameters, we fit the optimal models to the entire training set. Finally, we evaluate the model performance using the hold-out test set. To compare models we use the miss-classification rate, the AUC measure and the F1-score. We find that the benchmark model (Naive Bayes) has an accuracy of 63% while the Logistic Regression (linear model) has a better predictive power with accuracy of 82.7%. Nevertheless, the model that best fits the data is Random Forest, with a test accuracy of 88.3%, followed closely by Neural Network with a 87% test accuracy. These results suggest that the data has some non-linearities that are better captured by "black-box" flexible models. Specifically, we believe that Random Forest performs better than Neural Network given its ability to deal with categorical data, which is abundant in our data set.

The rest of the report is structured as follows. Section 2 presents the original data, exploratory data analysis, data cleaning and feature engineering strategies, and details the final data set used. Then, Section 3 succinctly presents each of the ML models selected to fit the data and the evaluation process. After this, Section 4 compares the predictive power of each model. Lastly, Section 5 presents our final remarks.

[1]Data set retrieve on January 18th 2023 at https://www.kaggle.com/data sets/jessemostipak/hotel-booking-demand.

2 Data

2.1 Original Data

The data set of choice is the *Hotel booking demand* data set available in Kaggle. The raw data set comprises of 32 columns and 119,390 rows, with 12 columns being categorical and the rest numerical. Each row represents a unique booking record, containing booking dates, lengths of stay, numbers of guests, booking platforms, meal plan types, and other information. Table 7 in the Appendix details each variable in the data set.

To explore the data, we calculated descriptive statistics and created visualizations to illustrate the structure and distributions of our variables. Four of the columns have missing values. The "children" column has four missing values, while the "country" column has 488 missing values. The "agent" column has 16,340 missing values, accounting for around 14% of the total rows. The "company" column has 112,593 missing values, indicating that the majority of rows lack company information.

It is important to note that some categorical variables in the data have an excessively large number of classes or categories. One-hot encoding these variables would result in an unwieldy and expensive data set with an unmanageable number of dimensions. Table 1 shows the number of categories in each variable.

Table 1: Number of Categories per Variable

Categorical Variable	**# of Categories**
arrival_date_month	12
assigned_room_type	12
country	177
customer_type	4
deposit_type	3
distribution_channel	5
hotel	2
market_segment	8
meal	5
reservation_status	3
reservation_status_date	926
reserved_room_type	10

2.2 EDA: Main Findings

Upon cleaning and one-hot encoding the raw data, we have the following findings:

1. We didn't find strong correlations between the variables (see Figure 22 in Appendix).
2. The percentage of repeated guests that cancel reservations (17%) is considerably lower than the percentage of first-time customers that end up canceling the hotel booking (61%).
3. For guests that opt for a non-refundable booking, there is a surprisingly high proportion of cancellations in comparison to non cancellations. This phenomenon appears to challenge the notion that the absence of a refund would generally discourage cancellations. (See Table 8)
4. A large majority of the guests are of Portuguese nationality, which matches the general intuition considering the hotels are based in Portugal. Furthermore, other common countries of origin are mostly from neighboring countries including the UK, France, and Spain (see Figure 23).
5. Individual and family stays constitute a large majority of the data; on the other hand, contract and group reservations appear to be far less prominent. (see Figure 24)
6. Bookings with a considerably large number of adults all appear to have been canceled. The graph also hints at potential outliers in the count of adults included in the reservation (see Figure 25).
7. The number of days a reservation is held in the waiting list is typically not too long and appears reasonable from a transactional perspective. However, there are instances where the number of days a reservation was held in the waiting list prior to confirmation was considerably long (see Figure 26)

8. Aggregating the bookings by arrival date, a clear pattern of seasonality can be observed from the data: the demand for hotel bookings surges during the summers and winters (see Figure 27 and Figure 28).

9. The two duration of stays variables are highly right-skewed. From Table 9 we can see that there are some bookings with very large outliers, while the majority of the stays last between 1 and 5 days. If we bin the columns, we observe roughly 46 rows with stays greater than 20 days.

10. There seems to be no significant variation in the average number of stays and the guests' demographics between reservations that were canceled and those that were not (see Table 10).

A complete EDA can be found in the adjunct Jupyter notebooks.

2.3 Data Cleaning, Transformations & Feature Engineering

After conducting our initial exploratory data analysis (EDA), we identified necessary transformations and feature engineering steps to prepare our data for analysis.

We removed a few bookings with negative adr values or values exceeding 5400, which appeared to be misentries, and replaced missing values in the children column with 0. The features lead_time, days_in_waiting_list and adr had right-skewed distributions, so we log-transformed them to ensure their distributions were almost normal in nature. Arrival_date_month was modified into the column arrival_month by substituting the month names with the corresponding month number.

A few new columns were created by combining groups within categorical columns or by combining two or more columns. The column domestic was created by assigning the value 'domestic' if the country of origin was Portugal and 'international' otherwise. We defined the binary feature got_room_booked by assigning values 1 or 0 depending on whether reserved_room_type was equal to assigned_room_type or not. Similarly, a new feature required_car_parking was assigned values 1 or 0 depending on whether the required_car_parking_spaces were more than zero. The features booked_by_agent and booked_by_company were created conditional on whether the columns agent or company were missing. Total_nights column was created by summing stays_in_weekend_nights and stays_in_week_nights. Similarly, previous_bookings was created by summing previous_cancellations and previous_bookings_not_cancelled.

Since the data set contained ISO3166 codes to represent countries, we used a mapping file to map the codes to a country and to a continent.[2] We only used the continent as a categorical variable for our analysis as countries would have too many classes.

A subset of the features from the original data set were removed from our analysis. company, country, agent, required_car_parking_spaces, stays_in_week_nights, stays_in_weekend_nights, previous_bookings_not_canceled, reserved_room_type, assigned_room_type were removed since we created new columns by grouping relevant values from these columns or by combining them with other columns. Also arrival_date_month, lead_time,adr, days_in_waiting_list were removed since we created new columns by transforming them. Other columns like arrival_date_week_number, arrival_date_day_of_month, distribution_channel, reservation_status and reservation_status_date were removed since they had too many factors or because they proved to be irrelevant for our analysis through the EDA.

The columns arrival_date_year, adults, children, babies, meal, market_segment, is_repeated_guest, booking_changes, deposit_type, customer_type, total_of_special_requests were left unchanged.

We created two additional versions of this transformed data set. In the first version, the categorical variables hotel, arrival_month,meal, market_segment, continent, is_repeated_guest, domestic, got_room_booked, booked_by_agent, customer_type, booked_by_company and required_car_parking were one hot encoded and in the second version these variables were dummy encoded by removing one category from each of them. Depending on the model to be fit, the appropriate data set was used.

2.4 Final Data Set Description

Table 2 describes the final data set after cleaning and transformations. Each row in the data set represents:

[2]Mapping file available at https://github.com/lukes/ISO-3166-Countries-with-Regional-Codes/blob/master/all/all.csv

Table 2: Data Description

Variable	Date Type	Description
adults	*Integer*	Number of adults
arrival_date_year	*Integer*	Year of arrival date
arrival_month	*Categorical*	Month of arrival date with 12 categories: 1 to 12
babies	*Integer*	Number of babies
booked_by_agent	*Categorical*	Whether the booking was made by an agent with categories: "Yes" and "No"
booked_by_company	*Categorical*	Whether the booking was made by a company with categories: 0 and 1
booking_changes	*Integer*	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
children	*Integer*	Number of children
continent	*Categorical*	Continent of origin with categories: "Africa", "Americas", "Antarctica","Asia","Europe","Oceania","Unknown"
customer_type	*Categorical*	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
deposit_type	*Categorical*	If the customer made a deposit to guarantee the booking with categories: No Deposit; Non Refund; Refundable
domestic	*Categorical*	If the customer is originally from Portugal ("domestic") or not ("international")
got_room_booked	*Categorical*	If the customer was assigned the roomtype he had originally reserved with categories: 1 and 0
hotel	*Categorical*	If the customer booked a City Hotel or a Resort Hotel
is_canceled	*Categorical*	Value indicating if the booking was BO canceled (1) or not (0)
is_repeated_guest	*Categorical*	Value indicating if the booking name was from a repeated guest (1) or not (0)
log_adr	*Float*	Logged value of the Average Daily Rate
log_days_in_waiting_list	*Float*	Logged value of the number of days before the booking was confirmed after booking
log_lead_time	*Float*	Logged value of the number of days that elapsed between the entering date of the booking into the PMS and the arrival date
market_segment	*Categorical*	Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"
meal	*Categorical*	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
previous_bookings	*Integer*	Total number of previous bookings made by the customer prior to the current booking
previous_cancellations	*Integer*	Number of previous bookings that were canceled by the customer prior to the current booking
required_car_parking	*Categorical*	Whether the customer requested for car parking spaces (1) or not (0)
total_nights	*Integer*	Total number of nights the guest stayed or booked to stay at the hotel
total_of_special_requests	*Integer*	Number of special requests made by the customer (e.g. twin bed or high floor)

3 Modelling

3.1 Train, Test & Cross Validation

In order to build and test the predictive power of different models, we randomly divided the data into train and test sets while ensuring that the proportion of cancellations was maintained the same in both. The train set consisted of 95,511 observations, while the test set had a total of 23,877 observations. The indexes for each test were stored, allowing the model to be run independently with the training set while using the same test set for evaluating the different models.

For each model, 10-fold cross-validation was used to determine the optimal hyperparameters (if needed). All models

were run in Python using the GridSearchCV method from the sklearn.model_selection module. This method iterates over different combinations of the selected parameters, performs cross-validation, and identifies the optimal combination based on evaluation metrics.

After determining the optimal model, we retrained it using the entire training set and assessed its predictive power using the test set. To evaluate the effectiveness of each model, we used accuracy, F-1 score and the AUC score.

3.2 Naive Bayes (benchmark)

For the Naive Bayes model, we used the 'hotel_bookings_ohe.csv' file and standardized the data after splitting it into train and test sets. We conducted grid searches for hyperparameter tuning using 10-fold cross-validation. The hyperparameter to be chosen was var_smoothing. By optimizing the misclassification rate, we found that the best model had a var_smoothing value of 0.01.

Using the optimized hyperparameter, we fitted the model to the entire training set in 0.106 seconds, which was the fastest model among the 9 models we analyzed.

After fitting the best model, we generated ALE plots for all of the numerical predictors to interpret the variable importance of each one (see Figure 1). Examining the ALE plots, we found that the top 5 most relevant numerical features are 'previous_cancellations', 'total_of_special_requests', 'total_nights', 'booking_changes', and 'adults'. The first-degree ALE plot showed that the effect of the predictor 'previous_cancellations' is positive and linear, the effect of 'total_nights' is negative and linear, the effect of 'total_of_special_requests' is negative and nonlinear, the effect of 'booking_changes' is negative and linear, and the effect of 'adults' is positive and almost linear.

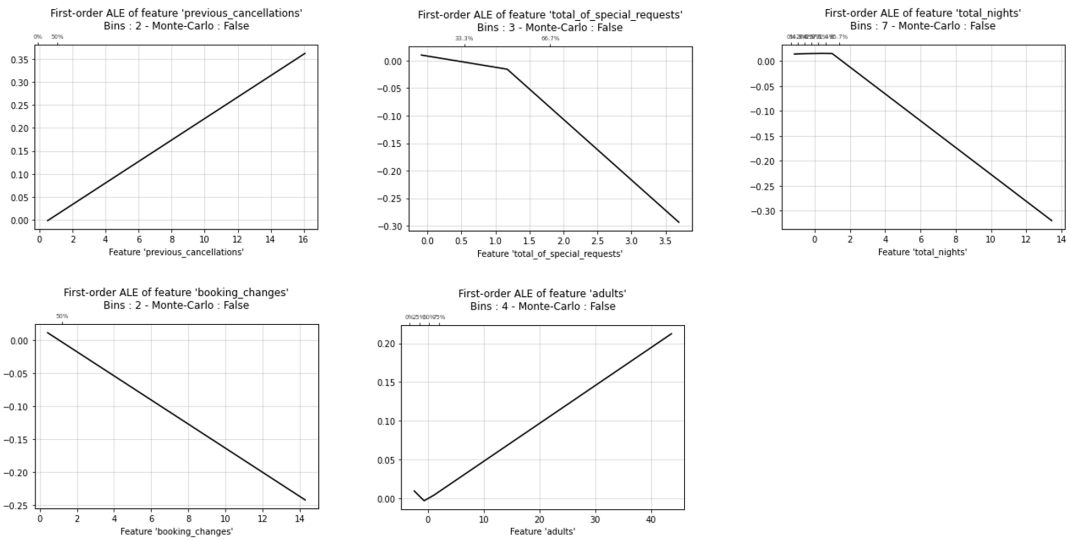

Figure 1: First-Degree Ale Plots for Optimal Naive Bayes Classifier

The second-degree ALE plots for the top 3 predictors in Figure 2 showed that there are interaction effects between 'previous_cancellations' and 'total_nights', 'total_nights' and 'total_of_special_requests', and 'trevious_cancellations' and 'total_of_special_requests'. Specifically, the effect of 'previous_cancellations' depends on 'total_nights', the effect of 'total_nights' depends on 'total_of_special_requests', and the effect of 'previous_cancellations' depends on 'total_of_special_requests'. However, the interaction effect between 'previous_cancellations' and 'total_of_special_requests' is much smaller than the main effects of each, indicating that particular interaction is not very significant.

After retraining the model using the complete training data set, we proceeded to make predictions on the unused test data set. Upon inputting the standardized test data, we achieved a test accuracy of 63% (equivalent to 37% misclassification rate), a test F-1 score of 69%, and a test AUC score of 68.8%.

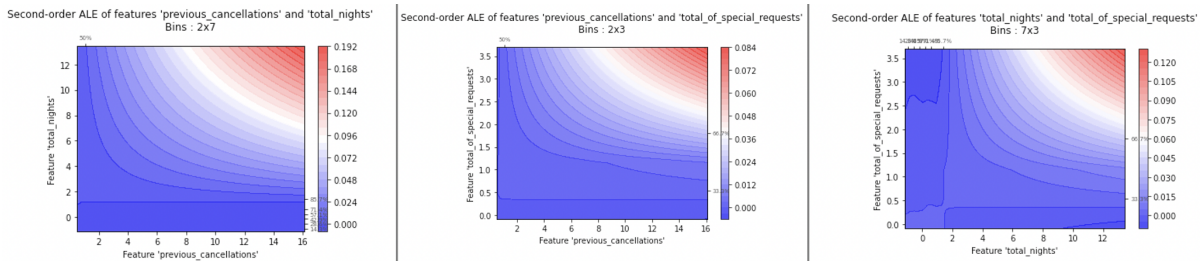

Figure 2: Second-Degree Ale Plots for Optimal Naive Bayes Classifier

3.3 Logistic Regression

As a first step, we fitted a logistic regression with regularization model to the "dummy" version of the data set (one less category for all categorical variables). We needed to remove the columns corresponding to *continent_Antarctica* and *market_segment_unknown* since only two of these rows had a value equal 1, which was causing estimation problems. We chose to use regularization here given the large number of features. After standardizing the data, we use 10-fold CV to tune the hyperparameters of the model: the regularization parameter as well as the type of penalty L1 (Lasso) or L2(Ridge). The corresponding tunning values tried were:

1. Regularization Parameter (*alpha*): 0.01, 0.1, 0.5, 1[3]
2. Penalty: L1, L2.

The best set of hyperparameters were $C = 10$ ($alpha = 0.1$) and $penalty = L2$ (Ridge Regression).

After this, we retrained the model using the entire training set. The training took 105.13 seconds. Table 3 shows the estimated coefficients, standard error, t-values and p-values of the estimates. Most of the coefficients are very small (given the regularization) and appear to be statistically significant, except *babies*, *previous_cancellations* and some categories of the categorical variables; however the last ones can not be removed from the model since the other categories are significant. It should be noted that, if any multicolinearity problems were present in the data, the Ridge regularization would have accounted for it.

To assess the variable importance, we ploted the main effects for all numerical variables. Figure 3 shows the main effects for the top 6 most important numerical variables based on the range of the y-axis (note that no main effects were plotted for categorical variables given the dummy version of the data set) which are number of previous cancellations (*previous_cancellations*), lead time of booking (*log_lead_time*), number of adults (*adults*), total number of special requests (*total_of_special_requests*), number of children (*children*) and total number of nights of stay (*total_nights*). Most of the important variables show a linear effect on the cancellations, except for the lead time booking, which is not surprising giving that logistic regression is a linear model, therefore would prioritize linear relations.Additionally, all the effects are positive except for number of special requests (i.e. the more requests, the less likely a customer is to cancel).

[3]The LogisticRegression function in Python uses the parameter C, the inverse of the regularization strength. This is $C = 1/alpha$.

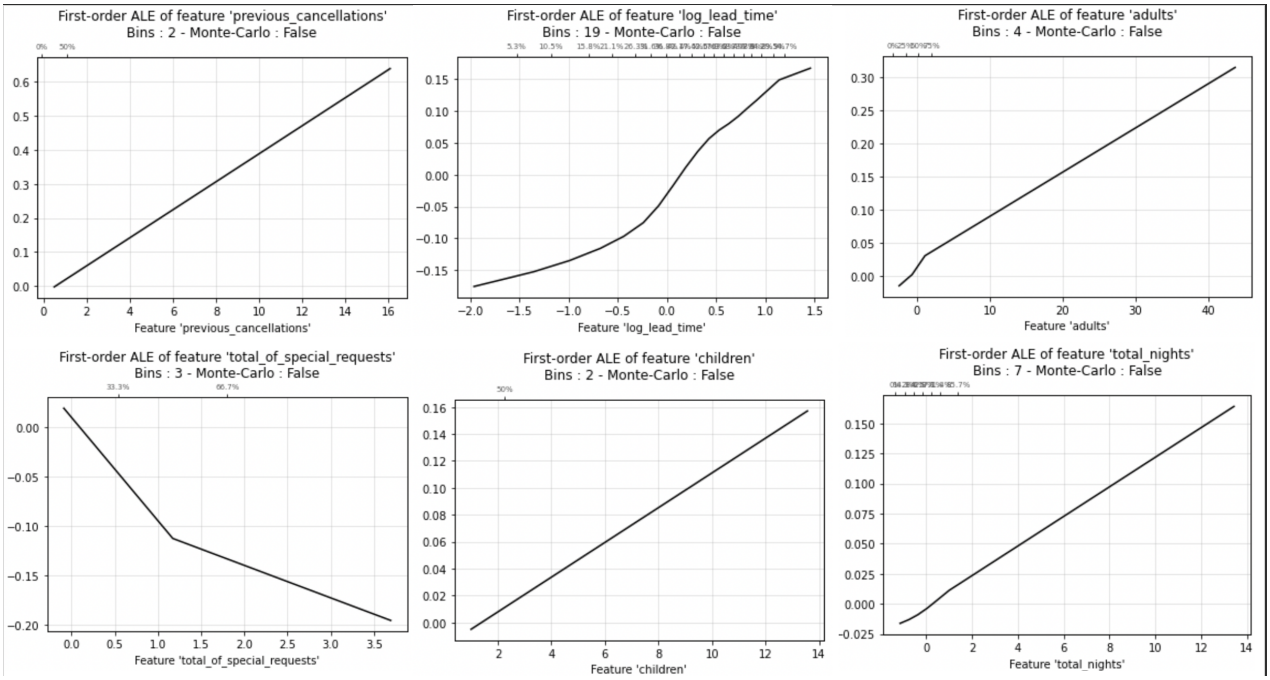

Figure 3: Main Effects for Top 6 Numerical Varaiables with Random Forest

Table 3: Logistic Regression with Regularization

Variable	Coefficient	Std Error	t-Value	p-value
intercept	0.370580	0.001182	313.449680	0.000000e+00
adults	0.013047	0.001261	10.342817	0.000000e+00
arrival_date_year	0.012114	0.001458	8.306857	2.220446e-16
arrival_month	0.005266	0.001374	3.833599	1.263627e-04
babies	0.000621	0.001192	0.520899	6.024385e-01
booked_by_agent	0.001589	0.001675	0.949059	3.425929e-01
booked_by_company	-0.010165	0.001868	-5.440838	5.316028e-08
booking_changes	-0.023985	0.001220	-19.664697	0.000000e+00
children	0.011721	0.001221	9.599923	0.000000e+00
continent_Americas	0.129946	0.001431	90.825120	0.000000e+00
continent_Asia	0.052032	0.001241	41.939896	0.000000e+00
continent_Europe	-0.007459	0.001371	-5.438615	5.382777e-08
continent_Oceania	-0.002667	0.001353	-1.970342	4.880212e-02
continent_unknown	0.019310	0.001451	13.305387	0.000000e+00
customer_type_Group	-0.003168	0.001230	-2.575837	1.000127e-02
customer_type_Transient	0.091732	0.001486	61.736363	0.000000e+00
customer_type_Transient-Party	-0.007016	0.001534	-4.572509	4.825215e-06
deposit_type_Non Refund	0.022708	0.001443	15.739054	0.000000e+00
deposit_type_Refundable	-0.053722	0.001231	-43.640793	0.000000e+00
domestic	0.013032	0.001318	9.887806	0.000000e+00
got_room_booked	-0.080558	0.001313	-61.333992	0.000000e+00
hotel	-0.008801	0.002086	-4.218031	2.466721e-05
is_repeated_guest	-0.004408	0.001988	-2.217690	2.657832e-02
log_adr	-0.029515	0.003002	-9.833206	0.000000e+00
log_days_in_waiting_list	-0.007409	0.001277	-5.800796	6.620843e-09
log_lead_time	0.007134	0.001289	5.533217	3.152325e-08
market_segment_Complementary	-0.002069	0.001226	-1.687989	9.141667e-02
market_segment_Corporate	0.027656	0.002527	10.944858	0.000000e+00
market_segment_Direct	-0.013570	0.002528	-5.368508	7.957535e-08
market_segment_Groups	0.122225	0.001662	73.558359	0.000000e+00
market_segment_Offline TA/TO	0.000120	0.001194	0.100206	9.201812e-01
market_segment_Online TA	0.003456	0.001505	2.295979	2.167921e-02
meal_FB	-0.008949	0.002790	-3.207656	1.338657e-03
meal_HB	-0.022512	0.004224	-5.329030	9.896198e-08
meal_SC	-0.005143	0.005713	-0.900141	3.680473e-01
meal_Undefined	-0.040566	0.006315	-6.423895	1.334526e-10
previous_bookings	0.052391	0.009169	5.713625	1.109223e-08
previous_cancellations	0.000503	0.001208	0.416528	6.770250e-01
required_car_parking	-0.004740	0.001260	-3.762636	1.682321e-04
total_nights	0.009229	0.001273	7.247466	4.278800e-13
total_of_special_requests	-0.009105	0.001211	-7.515931	5.706546e-14

Figure 4 shows the interaction (second order effects) for the top three numerical variables. As it can be observed, there are some interactions between all of them, although the range of the effect in this plots is somewhat smaller than the range of the main effects plots. For example, higher number of previous cancellations has a higher effect on the response if there is a smaller lead time booking than if the room was booked with a lot of time in advance.

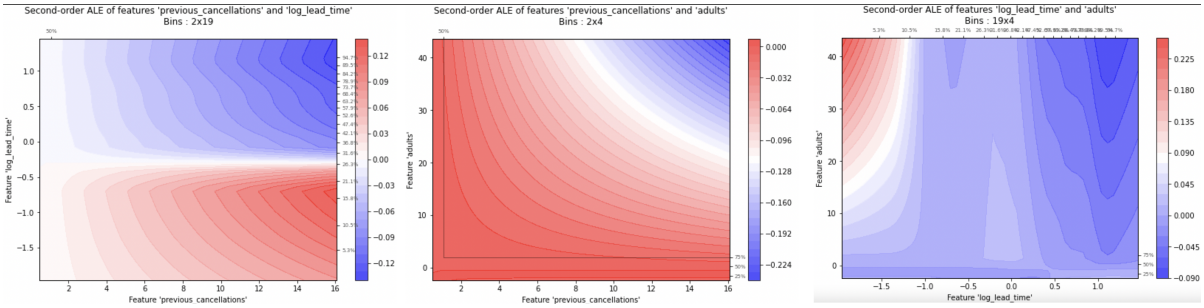

Figure 4: Second Order Effects for Top 3 Numerical Variables with Random Forest

Finally we used the test set to evaluate the model. Logistic regression with Ridge regularization achieves the following metrics: $miss - class\ rate = 0.173$, $accuracy = 0.827$, $F1\ score = 0.74$ and $AUC = 0.90$.

3.4 Neural Networks

The MLPClassifier from the sklearn.neural_network package was used to create the different neural network models. It creates a single layer perception with regularization. Before fitting a neural network model, the numerical features were standardized, as the magnitude of the features could affect the final weights. The hyperparameters which we considered for tuning are the regularization parameter alpha, the learning rate and the number of hidden layer neurons.

The first set of values considered for the parameters are alpha = [0.0001,0.001,0.1], learning_rate = [0.001,0.01,0.1] and hidden_layer_size = [10,20,30]. On performing cross validation, alpha=0.001,hidden_layer_size = 30 and learning_rate = 0.001 proved to be the best combination with a cross validation accuracy of 86.5%. Analyzing the performance of different combinations, we have noticed that models with higher number of neurons performed better, when the other parameters are kept constant. The learning rate did not have a significant impact on determining the accuracy.

Hence we ran a second set of hyperparameters for tuning with alpha = [0.0001,0.001,0.1] and hidden_layer_size = [30,40,50]. Learning_rate was fixed at the value of 0.001. As expected, the optimum model had hidden_layer_size = 50 and alpha=0.001 with a cross validation accuracy of 86.8%. Increasing the number of neurons thereafter did not showcase a significant improvement in accuracy. The final model was fit using the entire training data with a test accuracy of 87.2%. Figure 5 show the first order ALE plots.

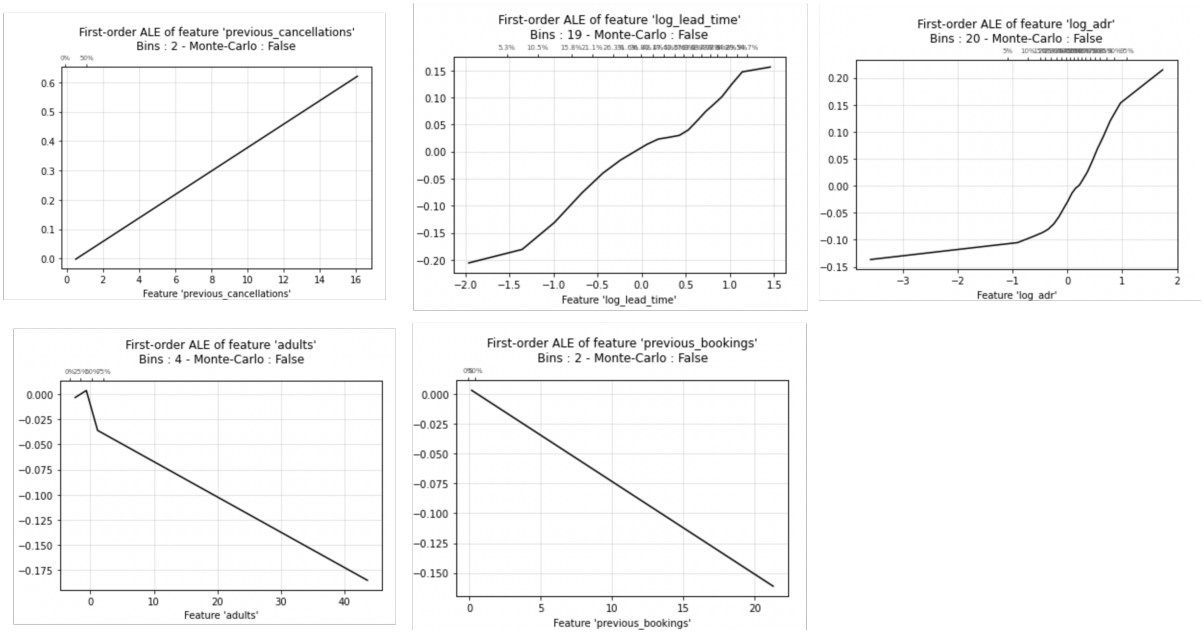

Figure 5: ALE plots for final neural network

Babies and log_days_in_waiting_list do not have a lot of distinct values and is mostly dominated by a single value which is 0. So they do not produce an ALE plot. Children,arrival_date_year,booking_changes and total_nights do not seem to have a significant effect on the target based on the range of values present on the ALE plot values.

The top 5 features which influence the values are as follows: i) previous_cancellations, ii) log_lead_time, iii) log_adr, iv) adults, and v) previous_bookings. Considering the top 5 features, previous_cancellations,log_adr and previous_bookings have a linear effect on the outcome. previous_cancellations have an increasing effect on the probability of cancellations while adults and previous_bookings have a decreasing effect on the probability. log_lead_time and log_adr have a non-linear increasing effect on the probability. The effect of log_adr almost seems quadratic. We can infer that maybe someone who has cancelled a lot previously have a higher probability to cancel again. A booking with a higher lead time could also have a higher probability of cancelling as there is a higher chance of their plans to change due to a large window of time between booking and arrival.

In Figure 6 we can also check for interactions using the 2 variable ALE plots. Based on the ranges of values in the interaction plots, we can conclude that a significant interaction effect is not present among previous_cancellation and log_lead_time and between previous_cancellations and log_adr. There is a noticeable interaction effect between log_adr and log_lead_time. For a fixed value of log_adr, the probability of cancelling decreases with increasing value of log_lead_time. This effect is more pronounced at high values of log_adr compared to low values.

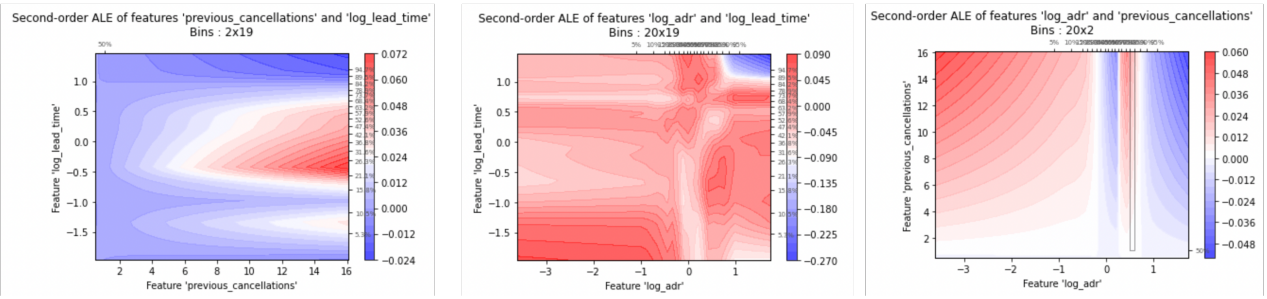

Figure 6: ALE plots for final neural network

3.5 Tree

For the Decision Tree model, we utilized the 'hotel_bookings_ohe.csv' data set. As decision tree models are not sensitive to the scale of the data, we did not standardize it. Through 10-fold cross-validation, we tuned the hyperparameters 'ccp_alpha' and 'max_depth' using 'criterion' as 'entropy' and found the optimal values to be 0.001 and 15, respectively. We then fit the best decision tree model to the entire training set.

To interpret the best model, we looked at the top 5 most important variables (see Figure 7), which were 'deposit_type_Non_Refund', 'market_segment_Online TA', 'domestic', 'required_car_parking 1', and 'total_of_special_requests'.

The first-degree ALE plots in Figure 8 indicated that the effect of 'total_of_special_requests' is negative and non-linear, the effect of 'log_lead_time' is negative and non-linear, the effect of 'arrival_date_year' is positive and linear, the effect of 'previous_cancellations' is positive and linear, and the effect of 'log_adr' is positive and non-linear.

The second-degree ALE plots in Figure 9 for 'log_lead_time', 'arrival_date_year', and 'total_of_special_requests' revealed significant interaction effects among them, indicating that the effect of each variable on the cancellation probability depends on one another. We can observe an interesting effect between log_lead_time and arrival_date_year. At lower values of log_lead_time, the arrival_date_year has a decreasing effect on the probability value, while at higher values, it has an increasing effect. The interactions between the other two sets of predictors are not much pronounced.

Using the test data set, we made predictions and calculated the performance metrics for our decision tree model. The results showed an accuracy of 84%, an F-1 score of 77%, and an AUC score of 81.9%.

3.6 Nearest Neighboors

The data set "hotel_bookings_dummy" was used for the K-Nearest Neighbor model, and we standardized the data. To determine the optimal value of k, we performed 5-fold cross-validation on the training data. A grid search was

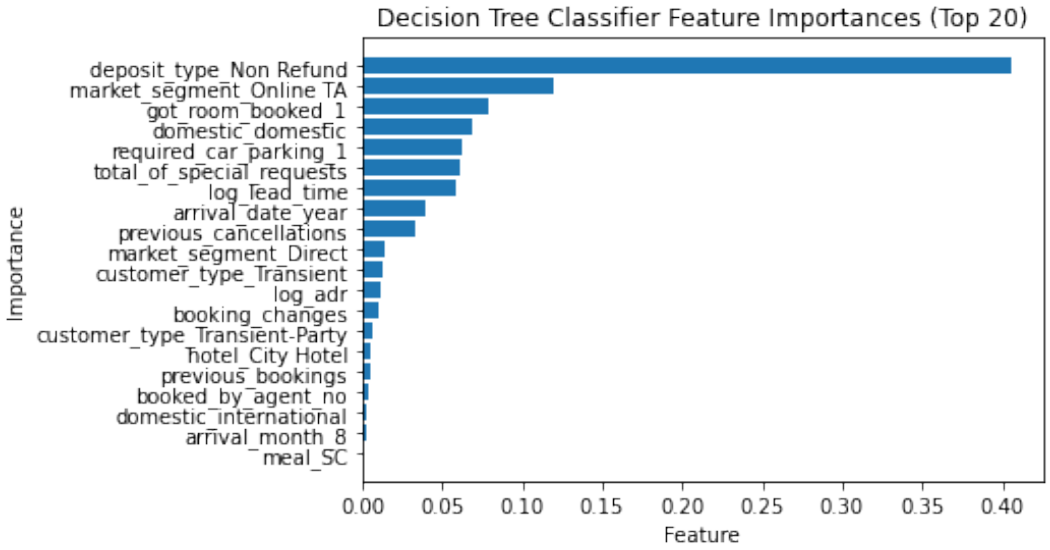

Figure 7: Variable Importance Plot for Decision Tree

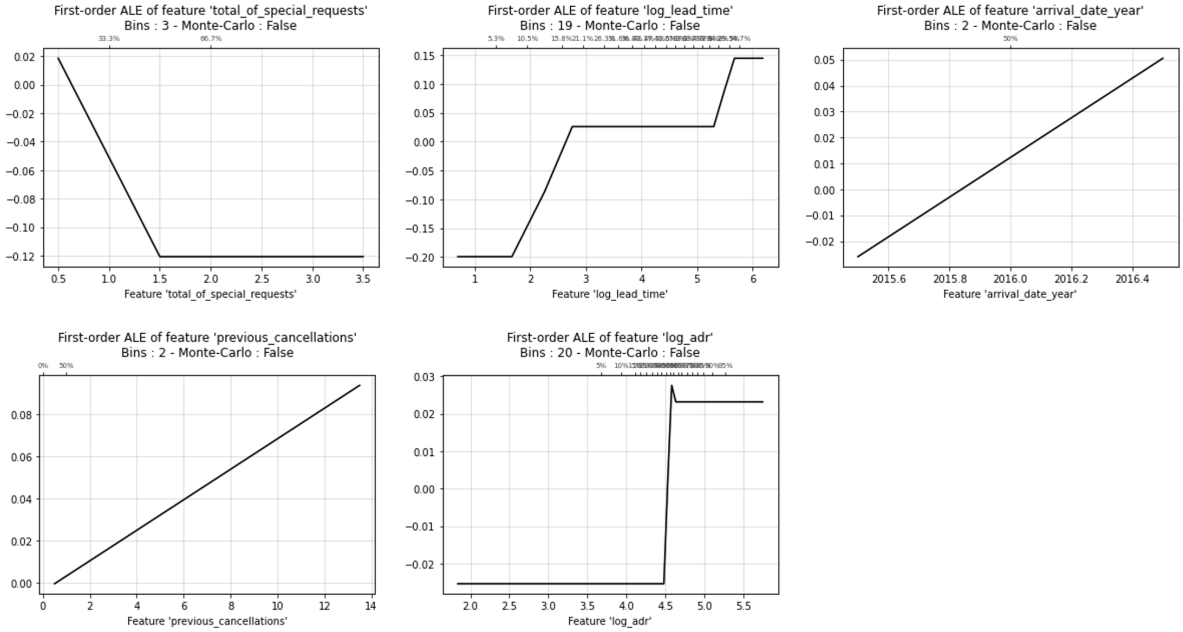

Figure 8: First-Degree ALE Plots for Decision Tree

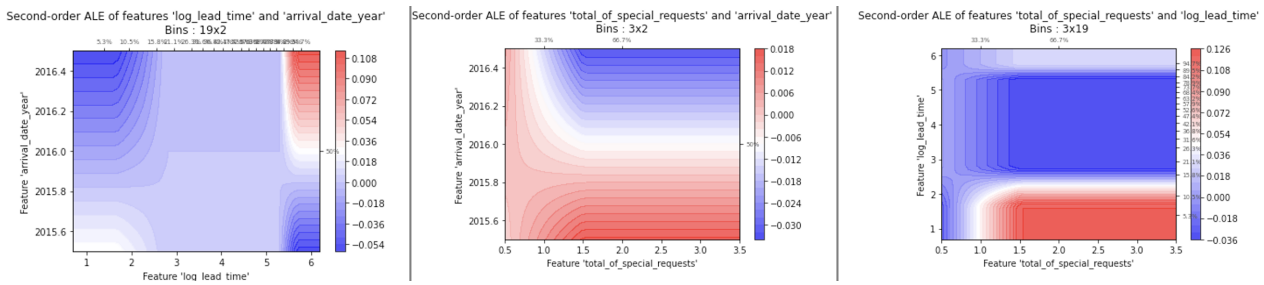

Figure 9: Second-Degree ALE Plots for Decision Tree

conducted on values of k ranging from 1 to 40, and the KNN model achieved the highest cross-validation accuracy (lowest misclassification rate) with k = 6 (see Figure 10).

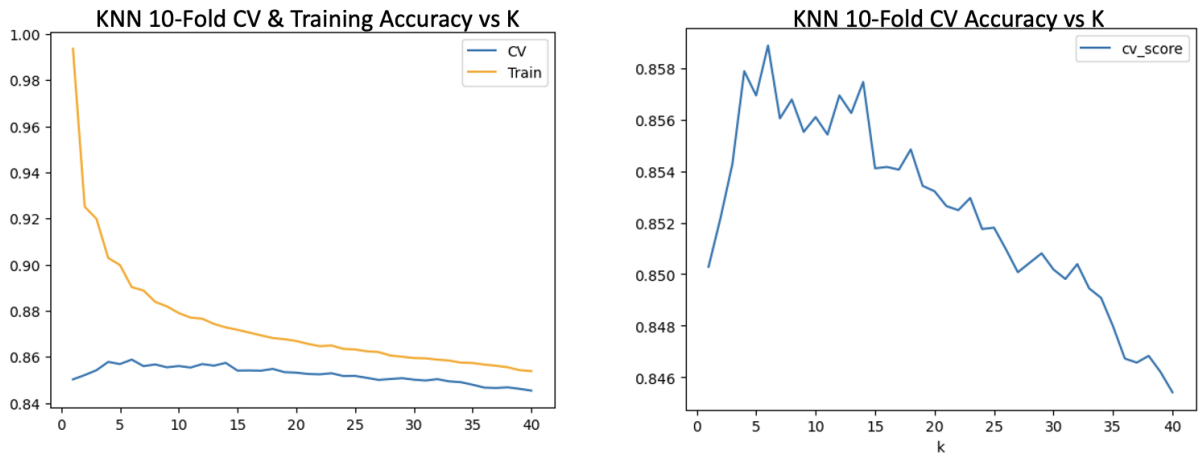

Figure 10: K-NN Selection

Once we determined the optimal value of k, we proceeded to find the six nearest neighbors in the training set for each row in the test set predictors. We then calculated the predicted class for the test set. Since KNN is a truly non-parametric method, there was no model to train. We timed the process of predicting the class label for the test set and noted the duration.

From analyzing the ALE plots, see Figure 11, we identified the five most important numeric variables as the number of adults, previous cancellations, lead time, number of special requests, and number of nights staying at the hotel. The predictor "adults" had a positive linear effect, "previous_cancellations" had a positive linear effect, "log_lead_time" had an almost linear and positive effect, "total_of_special_requests" had a non-linear and negative effect, and "total_nights" had a non-linear and somewhat quadratic effect (positive at first, then negative).

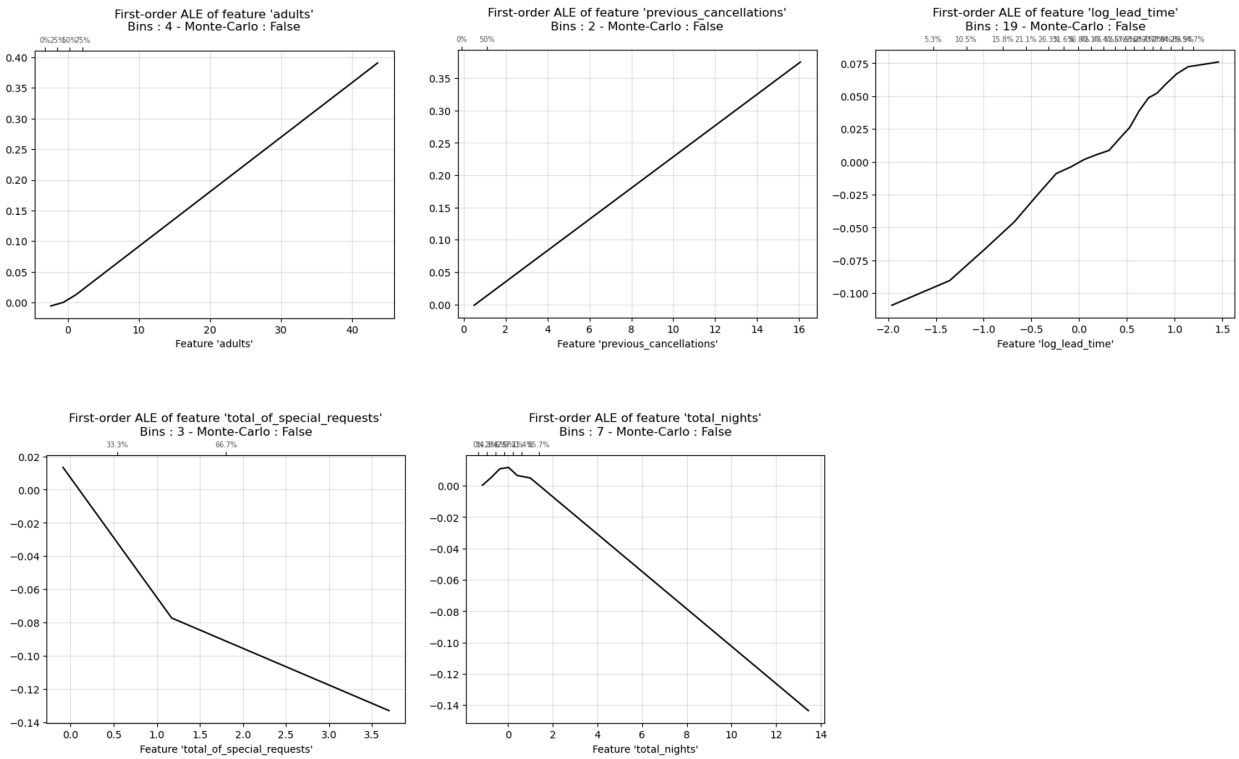

Figure 11: Main Effects with ALE plots for K-NN

Next Figure 12 shows the generated second-degree ALE plots for the top three most relevant numerical variables. These plots showed that the effect of 'adults' was dependent on 'previous_cancellations' and 'log_lead_time', while the effect of 'log_lead_time' was also dependent on 'previous_cancellations'. The ALE plots showed that

the interaction effect between 'previous_cancellations' and 'log_lead_time' is actually not very significant, but the other two interaction effects are. Interestingly, it seems at higher values of adults, extremely low and high values of lead_time increases the probability of cancellation while intermediate values decreases the probability.

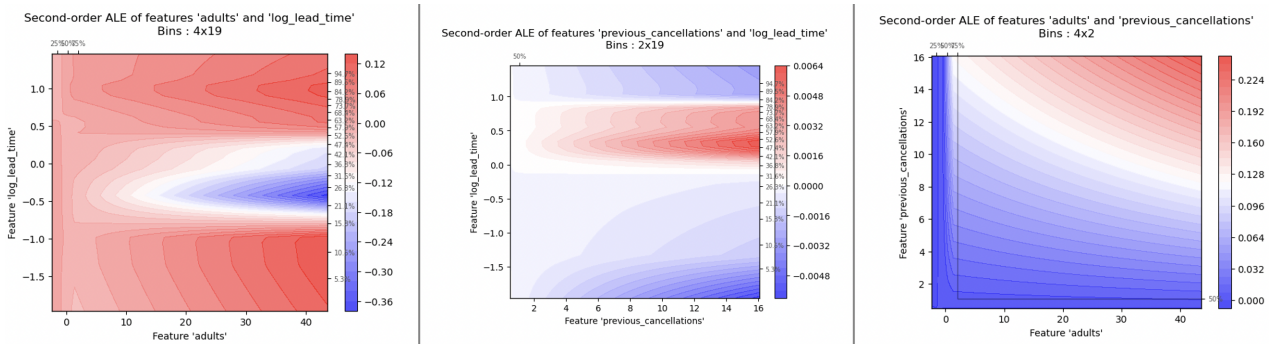

Figure 12: 2nd Order Effects with K-NN

After comparing the predicted cancellation label class with the actual label in the test set, we found that the best KNN model with k = 6 achieved an accuracy of 86.1%, an AUC score of 83.8%, and an F-1 score of 80.0%.

3.7 GAM

Before we try fitting a GAM model, we have to convert all categorical variables into factors. The variables children, babies and market segment had too many factors due to which the GAM model was not able to fit the data on cross validation and test data set. Hence they were not considered for our analysis. Even though the feature arrival_date_year should be considered as a continuous variable, it could not be introduced as one, due to less number of distinct values in the data. Hence it was also not considered in the GAM model. The variables adults, previous_cancellations, booking_changes, log_lead_time, total_nights, previous_bookings, log_days_in_waiting_list and log_adr are to be treated as smoothed features in the GAM model. The feature values were standardized before introducing the data set into the model.

Table 11 in Appendix shows the estimated coefficients for the parametric part of the model. We can notice that at least one category of every categorical feature is significant except for *required_car_parking*. We re-estimated the model removing this variable. Table 4 shows the final estimates, achieving a test set accuracy of 81.3%. The GAM model was computationally intensive to fit, taking almost 5 minutes to complete the training process.

Figure 13 presents the component graphs of the obtained GAM model. The effect of adults on the probability of cancellation is almost constant but the confidence interval of the effect gets wider as the number of adults increases. This implies the effect of adults is ambiguous. The same can be observed for the predictor previous_cancellations. A periodic pattern is also revealed on the impact of predictors such as booking_changes, previous_bookings, and log_adr on the probability of cancellation, where high values of these predictors alternate between increasing and decreasing the probability of cancellation at regular intervals.log_lead_time reveals to have a constant increasing effect on the cancellation probability. A increase in total_nights seem to showcase a decreasing trend at higher values but we have less number of points at that range to come to a conclusion. From the range of values of the component graphs, we can observe that top 5 effective predictors are: i) previous_bookings, ii) log_lead_time, iii) log_adr, iv) total_nights, and v)log_days_in_waiting_list. Since GAM is an additive model on the component functions of each of the features, we do not expect the model to showcase interactive effects between the features.

Note: PPR and LOESS models cannot be used for classification and hence were not considered for our analysis.

3.8 Boosting (XGBoost)

For the XGBoost model, we used the 'hotel_bookings_ohe.csv' data set. As XGBoost is a decision tree-based model, we did not standardize the data. Through 10-fold cross-validation, we tuned the hyperparameters 'learning_rate', 'max_depth', and 'n_estimators' and found the best values to be 0.1, 7, and 200 respectively.

We then fit the best model to the entire training set and interpreted it using the variable importance measures built-in to the model shown in Figure 14. The top 5 variables are 'deposit_type_Non_Refund', 'required_car_parking 1", 'market_segment_Online TA', 'previous_cancellations' and 'domestic'. We visualized the importance of these

Table 4: GAM Model Estimated Coefficients

Variable	Estimate	Std. Error	z value	$Pr(>\|z\|)$
(Intercept)	-4.988e-01	1.529e-01	-3.263	0.00110 **
hotelResort Hotel	-6.275e-02	2.602e-02	-2.412	0.01588 *
mealFB	-4.798e-01	1.225e-01	-3.917	8.98e-05 ***
mealHB	-4.211e-01	3.133e-02	-13.438	< 2e-16 ***
mealSC	6.162e-01	2.979e-02	20.689	< 2e-16 ***
mealUndefined	-1.157e+00	1.217e-01	-9.504	< 2e-16 ***
is_repeated_guest1	-7.999e-01	1.360e-01	-5.882	4.05e-09 ***
deposit_typeNon Refund	4.125e+00	1.285e-01	32.110	< 2e-16 ***
deposit_typeRefundable	4.622e-01	2.647e-01	1.746	0.08081 .
customer_typeGroup	-3.991e-02	2.214e-01	-0.180	0.85690
customer_typeTransient	1.273e+00	6.986e-02	18.225	< 2e-16 ***
customer_typeTransient-Party	1.616e-01	7.338e-02	2.202	0.02765 *
total_of_special_requests1	-1.017e+00	2.158e-02	-47.140	< 2e-16 ***
total_of_special_requests2	-1.081e+00	3.023e-02	-35.755	< 2e-16 ***
total_of_special_requests3	-1.422e+00	6.696e-02	-21.237	< 2e-16 ***
total_of_special_requests4	-2.184e+00	2.154e-01	-10.137	< 2e-16 ***
total_of_special_requests5	-3.130e+00	1.045e+00	-2.995	0.00274 **
arrival_month2	-9.557e-03	5.699e-02	-0.168	0.86682
arrival_month3	-3.811e-01	5.561e-02	-6.853	7.24e-12 ***
arrival_month4	-3.539e-01	5.511e-02	-6.421	1.35e-10 ***
arrival_month5	-5.695e-01	5.548e-02	-10.263	< 2e-16 ***
arrival_month6	-7.421e-01	5.662e-02	-13.106	< 2e-16 ***
arrival_month7	-9.265e-01	5.604e-02	-16.534	< 2e-16 ***
arrival_month8	-7.100e-01	5.613e-02	-12.650	< 2e-16 ***
arrival_month9	-8.481e-01	5.866e-02	-14.457	< 2e-16 ***
arrival_month10	-5.013e-01	5.653e-02	-8.868	< 2e-16 ***
arrival_month11	-2.897e-01	6.060e-02	-4.780	1.76e-06 ***
arrival_month12	-1.885e-01	6.013e-02	-3.135	0.00172 **
domesticinternational	-1.730e+00	2.331e-02	-74.221	< 2e-16 ***
continentAmericas	-1.011e+00	9.444e-02	-10.708	< 2e-16 ***
continentAntarctica	-6.867e+01	6.711e+07	0.000	1.00000
continentAsia	-7.946e-01	9.488e-02	-8.374	< 2e-16 ***
continentEurope	-1.461e+00	8.694e-02	-16.807	< 2e-16 ***
continentOceania	-1.503e+00	1.562e-01	-9.626	< 2e-16 ***
continentunknown	-8.270e-01	2.702e-01	-3.061	0.00220 **
got_room_booked1	1.865e+00	4.615e-02	40.413	< 2e-16 ***
booked_by_agentyes	4.684e-01	4.374e-02	10.708	< 2e-16 ***
booked_by_company1	-1.495e-01	6.925e-02	-2.158	0.03089 *

variables through a variable importance plot and analyzed the Average Local Effects (ALE) plot and second-degree ALE plot for the numerical predictors.

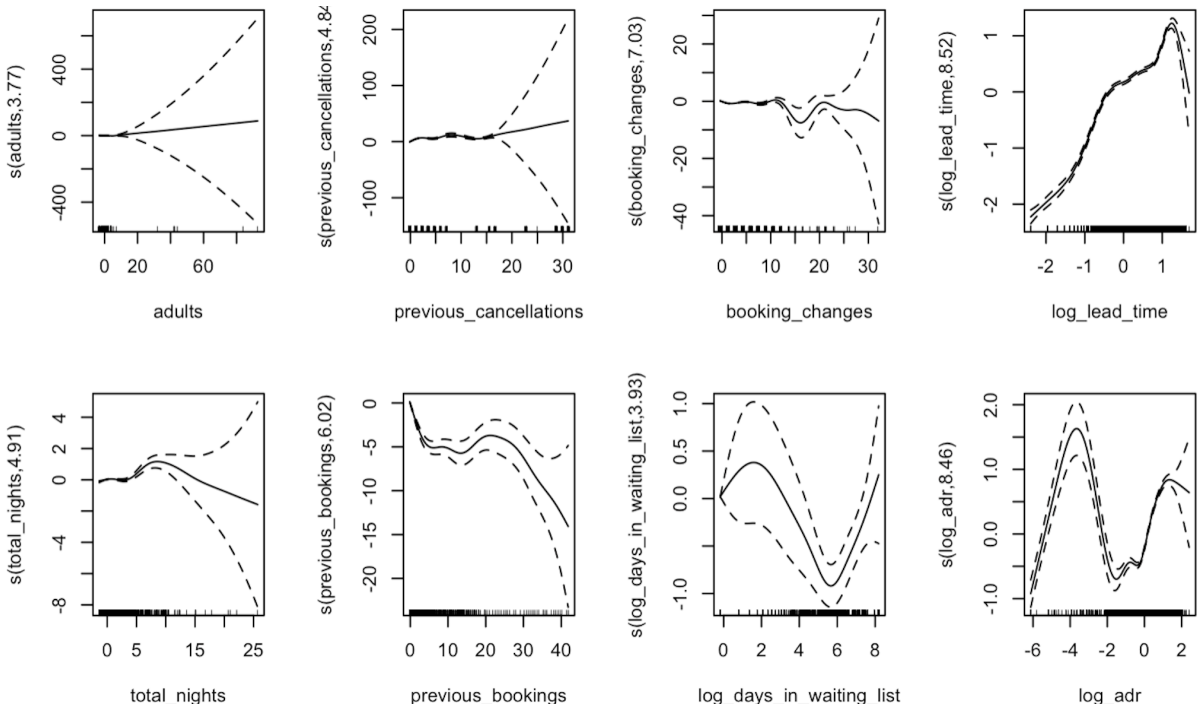

Figure 13: Component Graph

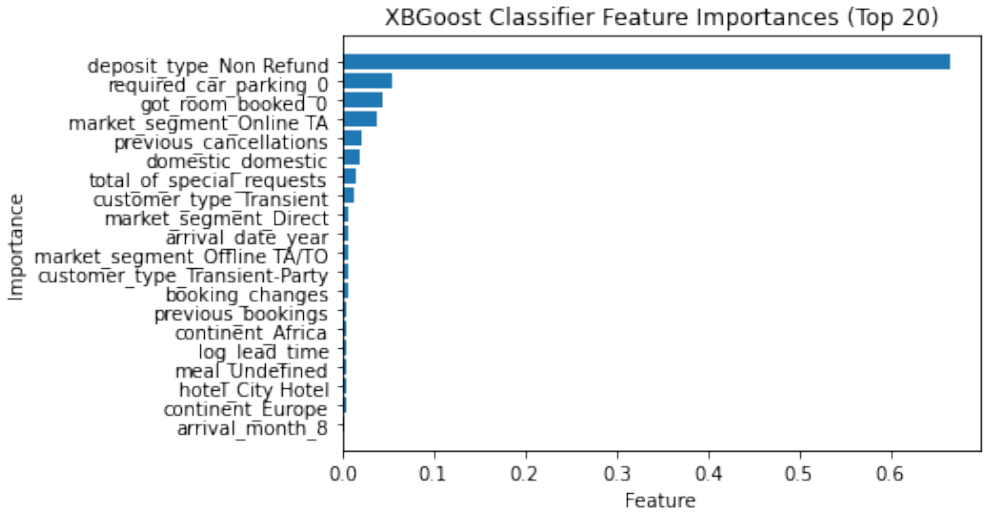

Figure 14: Variable Importance Plot for XGBoost

Figure 15 shows the first-degree ALE plot shows that the effect of the predictor 'previous_cancellations' is positive and linear, the effect of 'total_of_special_requests' is negative and nonlinear, the effect of 'booking_changes' is negative and linear, the effect of 'arrival_date_year' is positive and linear and the effect of previous_bookings is negative and linear.

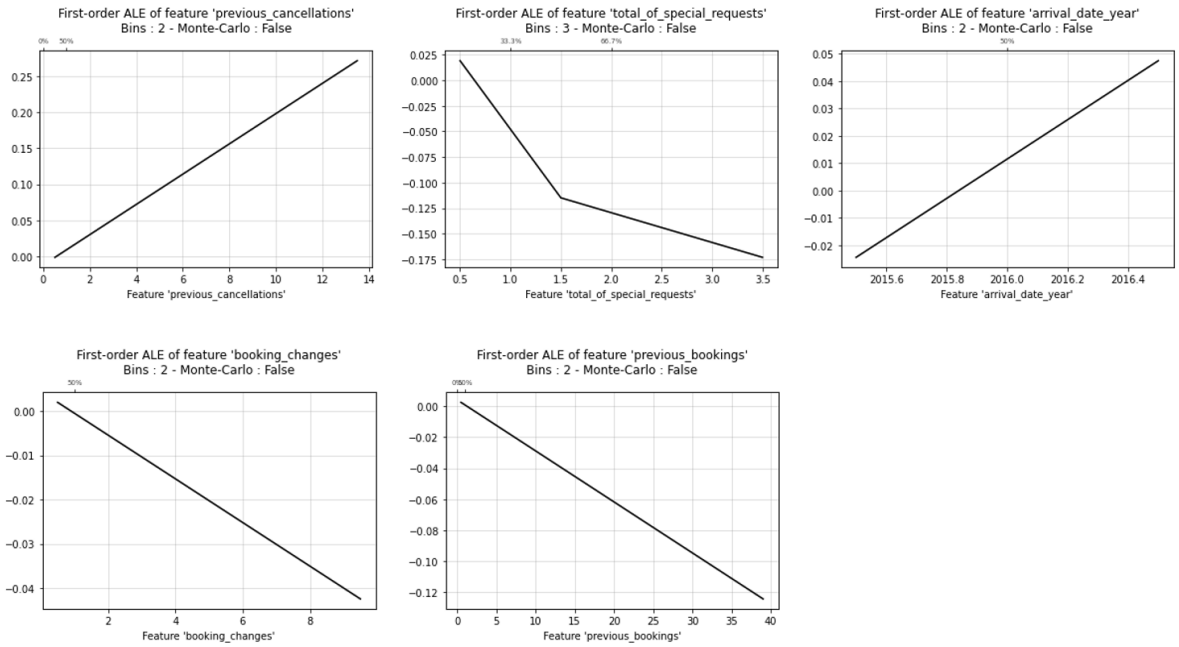

Figure 15: First-Degree ALE Plots for XGBoost

Upon examining the second degree ALE plots, Figure 16, we discovered interaction effects between 'previous_cancellations', 'arrival_date_year', and 'total_of_special_requests'. These predictors' effects on the cancellation probability are interdependent and rely on each other. However, the ALE plots also showed that the interaction effects between 'previous_cancellations' and 'special_requests' are insignificant comparing to the main effects of the two predictors.

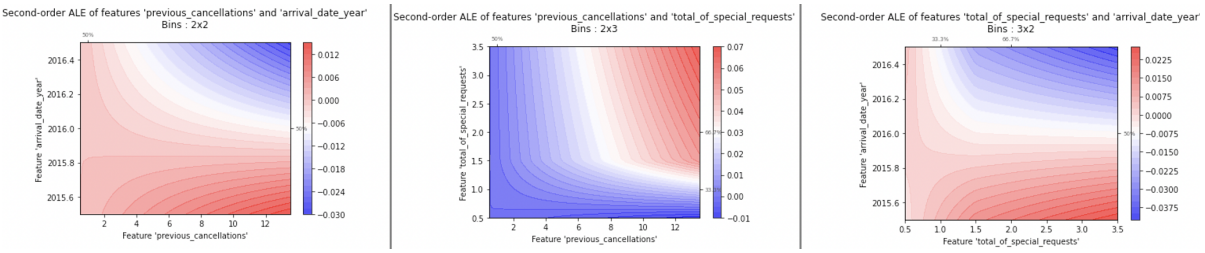

Figure 16: Second-Degree ALE Plots for XGBoost

Using the test data set, we made predictions and calculated the performance metrics for our model. The results showed a test accuracy of 87%, an F-1 score of 82%, and an AUC score of 85.7%.

3.9 Random Forest

We used the RandomForestClassifier function from the sklearn.ensemble package in Python to fit a classification Random Forest. For this model, the one-hot-encoding version of the data set was used (Random Forest does not need standardized data). First, we use 10-fold CV to tune the hyperparameters of the model. The parameters tuned and the corresponding values tried were:

1. Number of trees (*n_estimators*): 50, 100, 150, 200, 250, 500, 1000
2. Minimum number of samples in a leaf node (*min_samples_leaf*): 2,3,4,5
3. Number of features to consider for spliting (*max_features*): 1, 2, 3, 4, 5

The best set of hyperparameters were $n\_estimators = 1,000$, $min\_samples\_leaf = 2$ and $max\_features = 5$.

After this, we retrained the model using the entire training set. The training took 19.53 seconds. As for the variable importance, Figure 17 shows 20 most imortant features. Among those, the leading time of booking (*log_lead_time*), the type of deposit (*deposit_type*), the average daily rate (*log_adr*), the total number of requests (*total_of_special_requests*) and if it is a domestic customer (*domestic*) stand out.

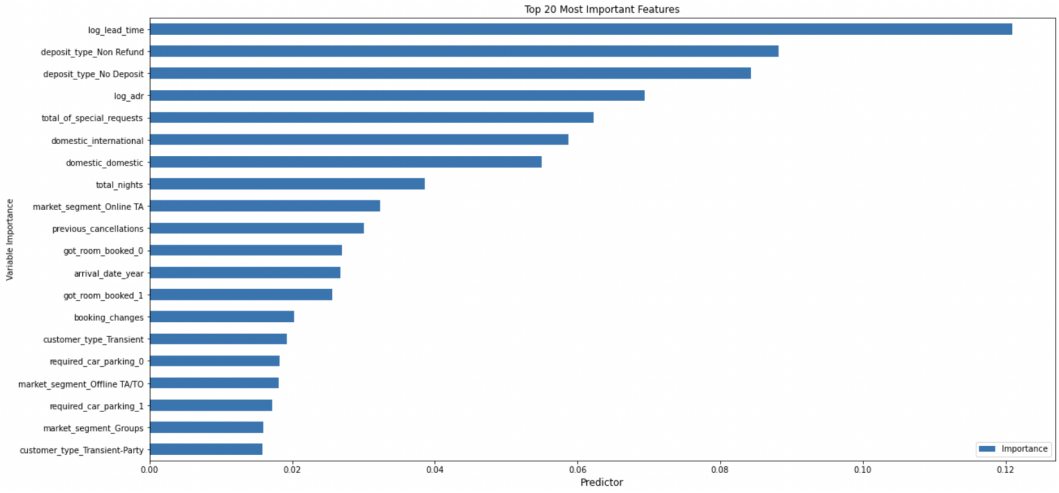

Figure 17: Variable Importance for Optimal Random Forest

Figure 18 shows the main effects for the top 6 most important numerical variables (note that no main effects were plotted for categorical variables given the ohe version of the data set). For the variables leading time of booking, the average daily price, the total number of special requests and total number of nights the effect is non-linear and it increases as the value of the variable increases, except for number of special requests where it decreases. Additionally, we can observe that the number of previous cancellations and the arrival date year have a linear positive effect in the response. This result is intuitive for the number of previous cancellations, if you normally cancel, there is a high chance that you will cancel again.

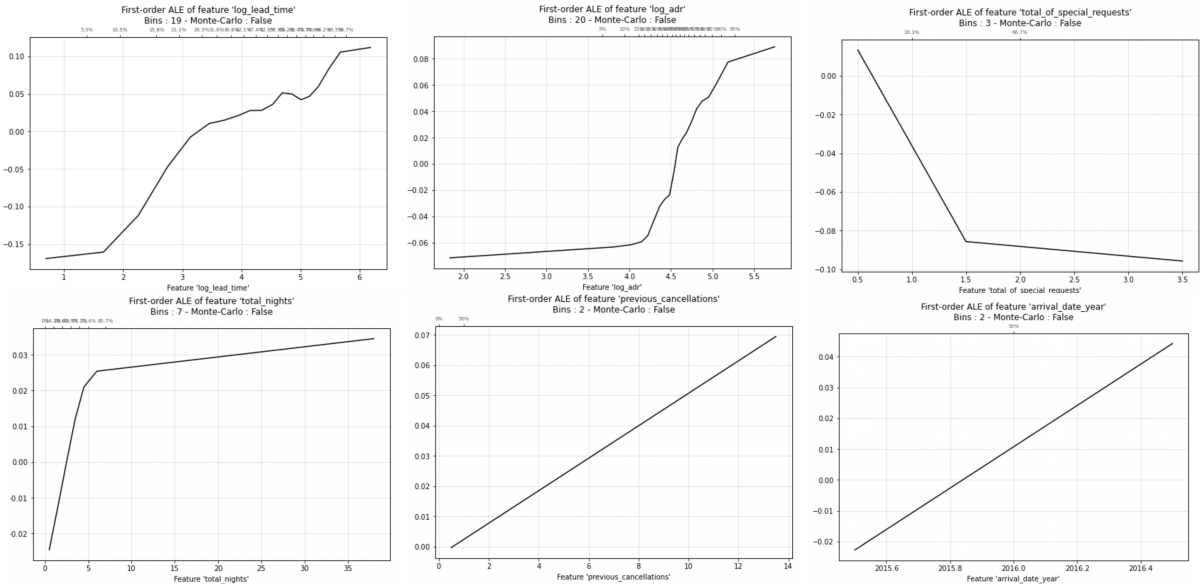

Figure 18: Main Effects for Top 6 Numerical Varaiables with Random Forest

Figure 19 shows the interaction (second order effects) for the top three variables. As can be observed, there are strong interactions between all of them. For example, higher values of lead time (booked with many time in advanced) at a small average daily rate has a higher impact on cancellations, while higher values of lead time but with higher average daily rate has a lower impact on cancellations. Similarly, a higher number of special requests has a different impact on cancellations depending on the lead time of booking and the average daily rate.

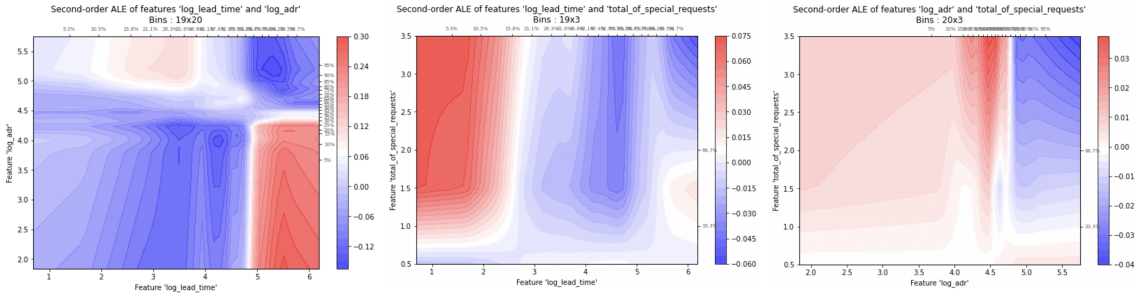

Figure 19: Second Order Effects for Top 3 Numerical Variables with Random Forest

Finally we used the test set to evaluate the model. Random Forest reaches the following metrics: $missclass\ rate = 0.1173$, $accuracy = 0.8827$, $F1\ score = 0.8320$ and $AUC = 0.95$.

3.10 Support Vector Machines

We used the data set "hotel_bookings_dummy" for the SVM model. Similar to KNN, we employed 10-fold cross-validation to tune the hyperparameters C, kernel type, gamma, and class_weight. Through cross-validation, we determined that the best set of parameters were $C = 0.1$, $class\_weight = None$, $gamma = 0.1$ and $kernel = rbf$. With these hyperparameters, we trained the SVM model with the entire training data set. The training process was computationally intensive, taking 59 seconds to complete.

To evaluate the importance of the numerical predictors, we generated first-degree ALE plots for each of them as shown in Figure 20. The plots revealed that the most important numerical predictors were 'log_lead_time', 'total_of_special_requests', 'adults', 'total_nights', and 'log_adr'. The variable "log_lead_time" exhibited a positive, nearly linear effect, while "total_of_special_requests" displayed a negative, linear effect. "Adults" had a non-linear, quadratic effect that was initially negative before becoming positive, "total_nights" had a positive, almost-linear effect, and "log_adr" had a somewhat quadratic, non-linear effect that was initially positive before becoming negative.

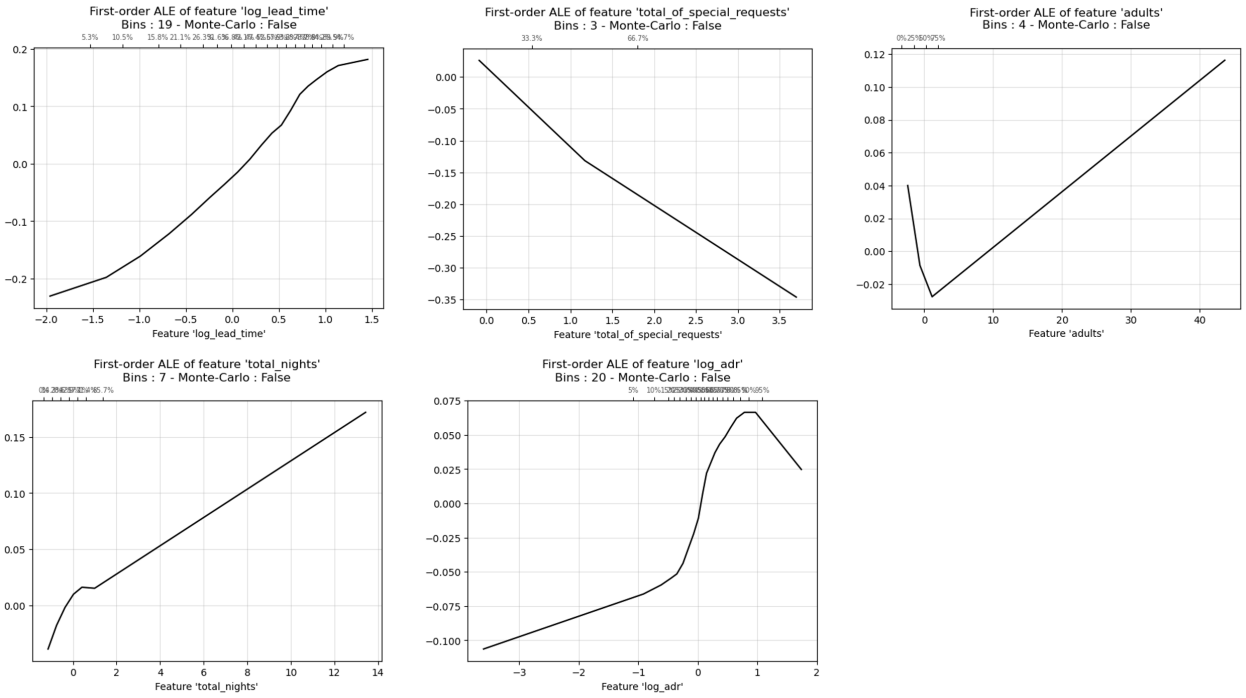

Figure 20: Main Effects With ALE Plots for SVM

According to the second degree ALE plot, see Figure 21, the impact of 'log_lead_time' varied depending on the values of 'adults' and 'total_of_special_requests'. Additionally, the effect of 'adults' was also dependent on the values of 'total_of_special_requests'.

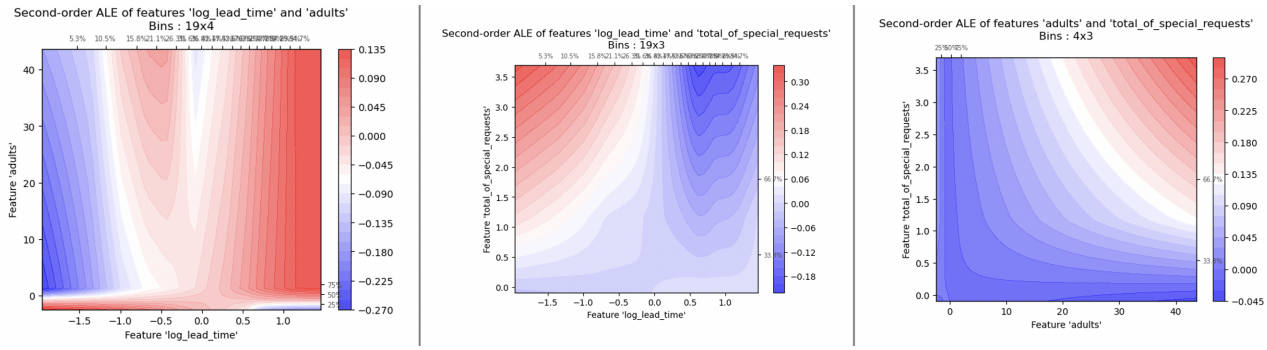

Figure 21: 2nd Order Effects With ALE Plots for SVM

In the end, we found that the best SVM model achieved a test accuracy of 56.4%, and an AUC score of 63.5%, and an F-1 score of 60.6%, making SVM the worst performing model in the entire exercise.

4 Model Comparison

Table 5 shows the wide range of performances across the 9 models fitted. Random Forest was the best performing model, with a test accuracy of 88.3%, followed closely by Neural Network and XGBoost, which achieved prediction test accuracy of over 87% each. These results were expected since the data set contained a large number of categorical variables, and tree-based models like Random Forest were better equipped to handle them. As expected, SVM was the worst performing model, and Naive Bayes also had lower prediction accuracy. However, the Naive Bayes model had a test accuracy 6.5 percentage points higher than that of the SVM model.

Table 5: Model Comparison for Test Set

Model	Accuracy	Missclass	F-1 Score	AUC	Training Time (s)	Top 5 Important Features
Random Forest	0.883	0.117	0.832	0.950	19.5	log lead time, log adr,total of special requests, total nights, previous cancellations
Neural Network	0.872	0.128	0.824	0.859	60.9	previous cancellations, log lead time, log adr, adults, previous bookings
XGBoost	0.870	0.130	0.820	0.857	6.0	Deposit type non refund, required car parking, market segment online, previous cancellations, domestic
KNN	0.861	0.139	0.800	0.838	3.6	adults, previous cancellations, log lead time, total of special requests, total nights
Decision Tree	0.840	0.160	0.770	0.819	0.5	Deposit type non refund, market segment online,domestic, required car parking, total of special requests
Logistic Regression	0.827	0.173	0.746	0.900	150.1	previous cancellations, log lead time, adults, total of special requests, children, total nights
GAM	0.813	0.187	0.719	0.779	297.6	previous bookings, log lead time, log adr, total nights, log days in waiting list
Naive Bayes	0.630	0.370	0.690	0.688	0.1	previous cancellations, total of special requests, total nights, booking changes, adults
SVM	0.564	0.436	0.606	0.635	59.4	log lead time, total of special requests, adults, total nights, log adr

Although class imbalance was not a huge issue, with roughly 36% of the label class being 1, the ROC AUC metric showed that the runner-up to Random Forest was actually the logistic regression model. Another dimension to consider was the computational expense of the models, indicated by the training time. Our results table revealed that the training time values varied dramatically. Naive Bayes and Decision Tree models took less than a second to train, while logistic regression took 150 seconds. Among the top performers, Random Forest took roughly 19 seconds to train, which was relatively fast, and XGBoost was even faster, delivering a test accuracy of 87% in less than 6 seconds. On the other hand, SVM was the worst performing model, taking almost 60 seconds to train, which exhibited an incredibly low ROI.

Based on Table 6, the most common important features for booking cancellations were the lead time of each booking, the total number of special requests, the average daily rate, the number of previous cancellations, and the number of adults. However, it is important to note that for some algorithms without a built-in method of importance, categorical features may not appear, as ALE plots used in the analysis do not plot anything for categorical variables.

Table 6: Most Important Features Accross Methods

Feature	No. Appearences
log_lead_time	6
total_of_special_requests	6
total_nights	6
previous_cancellations	5
adults	5
log_adr	4
previous_bookings	2
deposit_type	2
required_car_parking	2
market_segment	2
domestic	2
children	1
days_in_wait_list	1
number_booking_changes	1

5 Final Remarks

Through this project we applied supervised-learning classification models to study hotel booking cancellations. First, we performed an exploratory data analysis, cleaned and transformed the predictors and created three different versions of our data set: *clean* (numerical and categorical variables), *dummy* (numerical, each categorical feature with k categories is represented with k-1 dummy variables) and *ohe* (numerical, each categorical feature with k categories is represented with k dummy variables). Each model used a specific version of the data depending on its requirements and characteristics, and the data sets were randomly split into train and test, and the same splits were used across models. We then developed nine ML models to predict hotel booking cancellations, using Naive Bayes as our benchmark method. Specifically, we considered Naive Bayes classifier, Logistic Regression, Neural Network, Tree, k-Nearest Neighbors, GAM, XGBoost, Random Forest and Support Vector Machines, where SVM was an additional model not studied in class. We tuned and trained every model using cross validation for the training set, and assessed the predictive powers of the models using the test set. The performance metrics used were missclassification rates, accuracies, F1 Scores and AUC scores. After analyzing the data, we found that the Random Forest model was the best fit with a test accuracy of 88.3%. The Neural Network and XGBoost models closely followed, achieving test accuracies of over 87% each. Notably, XGBoost had the fastest training time among these three models, being 10 times faster than Neural Networks and 3 times faster than Random Forest. This could be a crucial consideration from a business standpoint, as accurate predictions may be required daily, with the need for model retraining every week or at a higher frequency for all hotels in the chain. Therefore, the trade-off of sacrificing a mere 0.013 accuracy to reduce training time to one third is worth it and can significantly impact hotel management and resource allocation, ultimately translating into more profit for the hotel.

6 Appendix

6.1 Original Data Set Description

Table 7: Data Description

Variable	Date Type	Description
ADR	*Numeric*	Average Daily Rate
Adults	*Integer*	Number of adults
Agents	*Categorical*	ID of the travel agency that made the booking
ArrivalDateDayOfMonth	*Integer*	Day of the month of the arrival date
ArrivalDateMonth	*Categorical*	Month of arrival date with 12 categories: "January" to "December"
ArrivalDateWeekNumber	*Integer*	Week number of the arrival date
ArrivalDateYear	*Integer*	Year of arrival date
AssignedRoomType	*Categorical*	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 (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
Babies	*Integer*	Number of babies
BookingChanges	*Integer*	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
Children	*Integer*	Number of children
Company	*Categorical*	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
Country	*Categorical*	Country of origin. ISO 3155-3:2013 format
CustomerType	*Categorical*	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
DaysInWaitingList	*Integer*	Number of days the booking was confirmed
DepositType	*Categorical*	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit; Non Refund; Refundable
DistributionChannel	*Categorical*	Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators"
IsCanceled	*Categorical*	Value indicating if the booking was BO canceled (1) or not (0)
IsRepeatedGuest	*Categorical*	Value indicating if the booking name was from a repeated guest (1) or not (0)
LeadTime	*Integer*	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
MarketSegment	*Categorical*	Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"
Meal	*Categorical*	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
PreviousBookingsNotCanceled	*Integer*	Number of previous bookings not canceled by the customer prior to the current booking
PreviousCancellations	*Integer*	Number of previous bookings that were canceled by the customer prior to the current booking
RequiredCardParkingSpaces	*Integer*	Number of car parking spaces required by the customer
ReservationStatus	*Categorical*	Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why
ReservationStatusDate	*Date*	Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel
ReservedRoomType	*Categorical*	Code of room type reserved. Code is presented instead of designation for anonymity reasons
StaysInWeekendNights	*Integer*	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
StaysInWeekNights	*Integer*	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
TotalOfSpecialRequests	*Integer*	Number of special requests made by the customer (e.g. twin bed or high floor)

6.2 Exploratory Data Analysis: Highlights

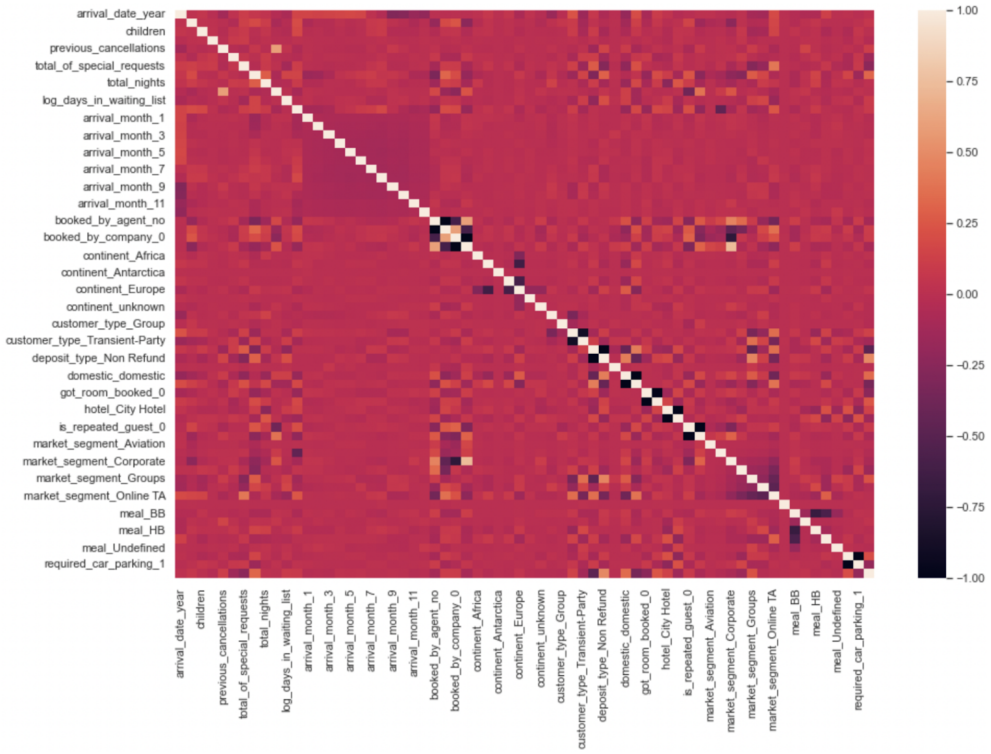

Figure 22: Correlation Plot

Table 8: Number of Categories per Variable

is_canceled	No Deposit	Non Refund	Refundable
0	74947	93	126
1	29694	14494	36

Table 9: Length of stay in days.

Class	Count
0-5	114537
6-10	4451
11-15	245
16.20	111
>20	46

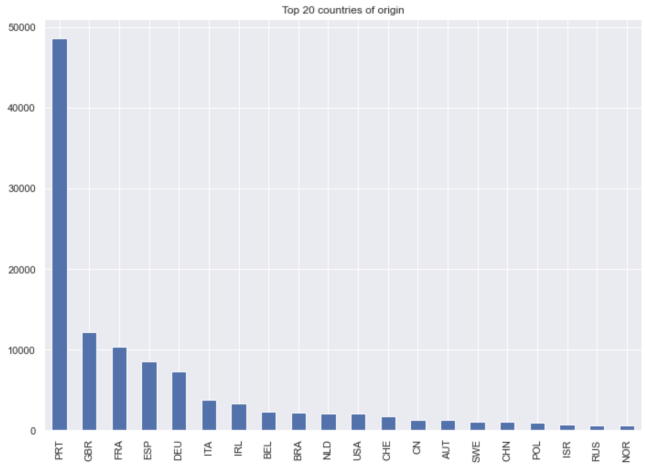

Figure 23: Top 20 Countries of Origin

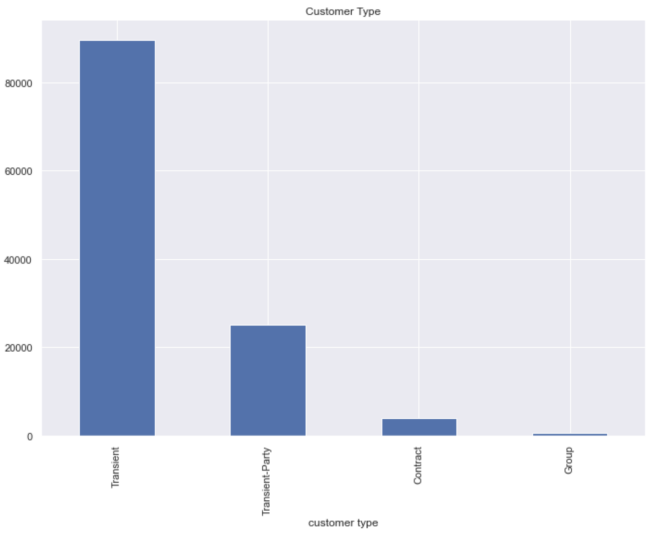

Figure 24: Customer Type Distribution

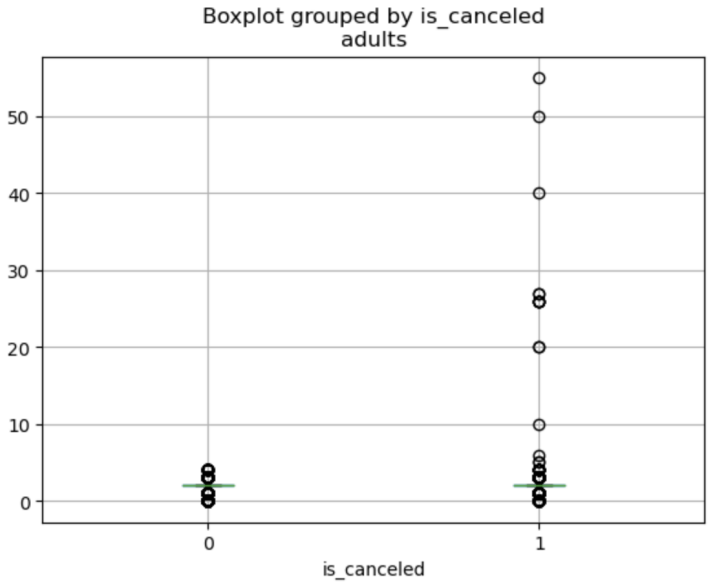

Figure 25: Adult distribution by cancellation.

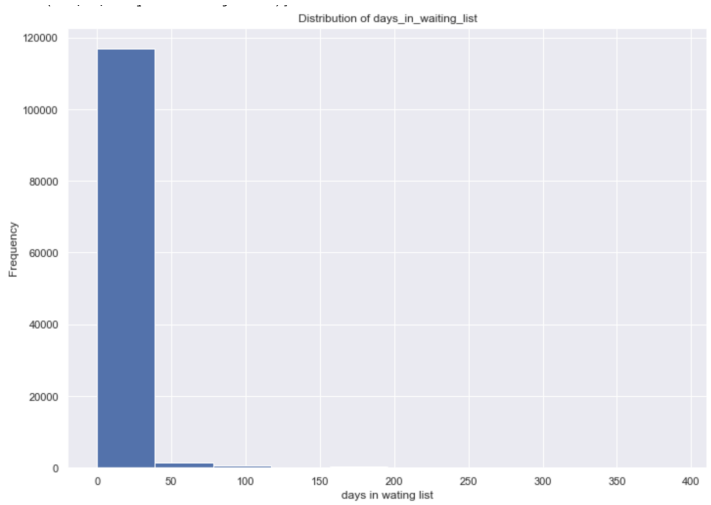

Figure 26: Days in waiting list.

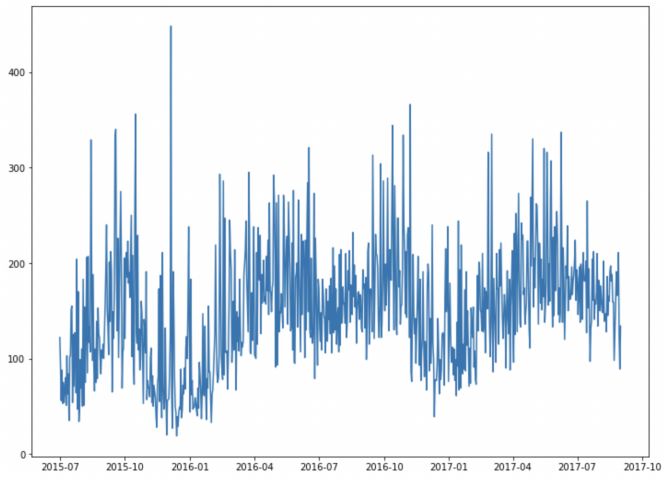

Figure 27: Number of Bookings for each Arrival Date.

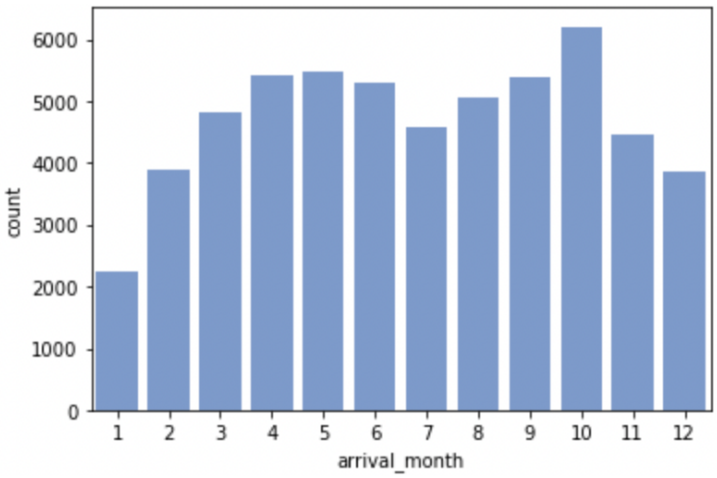

Figure 28: Number of Bookings for each Arrival Month

Table 10: Number of stays in week days and guest composition.

is_canceled	Stays in week nights	Adults	Children	Babies
0	2.4640	1.8297	0.1023	0.0103
1	2.5619	1.9017	0.1065	0.0038

6.3 GAM Model

Table 11: GAM Model Estimated Coefficients

Variable	Estimate	Std. Error	z value	$Pr(>\|z\|)$
(Intercept)	-3.323e-01	1.544e-01	-2.153	0.031353 *
hotelResort Hotel	1.730e-01	2.722e-02	6.358	2.05e-10 ***
mealFB	-4.573e-01	1.301e-01	-3.514	0.000441 ***
mealHB	-4.730e-01	3.236e-02	-14.616	< 2e-16 ***
mealSC	6.172e-01	3.012e-02	20.490	< 2e-16 ***
mealUndefined	-1.351e+00	1.244e-01	-10.856	< 2e-16 ***
is_repeated_guest1	-7.670e-01	1.385e-01	-5.538	3.06e-08 ***
deposit_typeNon Refund	4.016e+00	1.292e-01	31.077	< 2e-16 ***
deposit_typeRefundable	3.694e-01	2.703e-01	1.367	0.171720
customer_typeGroup	-4.575e-02	2.251e-01	-0.203	0.838936
customer_typeTransient	1.348e+00	7.062e-02	19.088	< 2e-16 ***
customer_typeTransient-Party	1.571e-01	7.416e-02	2.118	0.034151 *
total_of_special_requests1	-1.029e+00	2.213e-02	-46.491	< 2e-16 ***
total_of_special_requests2	-1.094e+00	3.100e-02	-35.285	< 2e-16 ***
total_of_special_requests3	-1.429e+00	6.856e-02	-20.847	< 2e-16 ***
total_of_special_requests4	-2.217e+00	2.180e-01	-10.170	< 2e-16 ***
total_of_special_requests5	-3.261e+00	1.059e+00	-3.078	0.002081 **
arrival_month2	-4.257e-02	5.803e-02	-0.733	0.463260
arrival_month3	-4.097e-01	5.662e-02	-7.235	4.64e-13 ***
arrival_month4	-4.117e-01	5.626e-02	-7.317	2.53e-13 ***
arrival_month5	-6.620e-01	5.662e-02	-11.691	< 2e-16 ***
arrival_month6	-8.053e-01	5.785e-02	-13.919	< 2e-16 ***
arrival_month7	-9.894e-01	5.728e-02	-17.274	< 2e-16 ***
arrival_month8	-7.780e-01	5.743e-02	-13.549	< 2e-16 ***
arrival_month9	-9.034e-01	5.988e-02	-15.087	< 2e-16 ***
arrival_month10	-5.542e-01	5.760e-02	-9.621	< 2e-16 ***
arrival_month11	-3.231e-01	6.165e-02	-5.241	1.60e-07 ***
arrival_month12	-2.016e-01	6.142e-02	-3.282	0.001030 **
domesticinternational	-1.827e+00	2.425e-02	-75.326	< 2e-16 ***
continentAmericas	-1.007e+00	9.638e-02	-10.451	< 2e-16 ***
continentAntarctica	-9.882e+01	6.711e+07	0.000	0.999999
continentAsia	-8.092e-01	9.681e-02	-8.358	< 2e-16 ***
continentEurope	-1.477e+00	8.873e-02	-16.646	< 2e-16 ***
continentOceania	-1.518e+00	1.584e-01	-9.583	< 2e-16 ***
continentunknown	-6.388e-01	2.834e-01	-2.254	0.024183 *
got_room_booked1	1.854e+00	4.669e-02	39.711	< 2e-16 ***
booked_by_agentyes	4.129e-01	4.517e-02 v 9.141	< 2e-16 ***	
booked_by_company1	-2.063e-01	7.064e-02	-2.920	0.003501 **
required_car_parking1	-8.766e+01	8.697e+05	0.000	0.999920

References

Antonio, Nuno, Ana de Almeida, and Luis Nunes. 2019. "Big Data in Hotel Revenue Management: Exploring Cancellation Drivers to Gain Insights into Booking Cancellation Behavior." *Cornell Hospitality Quarterly* 60 (4): 298–319.

Chen, Chih-Chien, Zvi Schwartz, and Patrick Vargas. 2011. "The Search for the Best Deal: How Hotel Cancellation Policies Affect the Search and Booking Decisions of Deal-Seeking Customers." *International Journal of Hospitality Management* 30 (1): 129–35.

Falk, Martin, and Markku Vieru. 2018. "Modelling the Cancellation Behaviour of Hotel Guests." *International Journal of Contemporary Hospitality Management* 30 (10): 3100–3116.

Hollander, Jordan. Accessed March 3rd 2023. "50+ Hospitality Statistics You Should Know (2023)." Hotel Tech Report; https://hoteltechreport.com/news/hospitality-statistics.

Loeb, Tony. Accessed March 3rd 2023. "Where Do Cancellations Come From?" Hotel Tech Report; https://hoteltechreport.com/news/hospitality-statistics.

Sanchez-Medina, Agustin J, C Eleazar, et al. 2020. "Using Machine Learning and Big Data for Efficient Forecasting of Hotel Booking Cancellations." *International Journal of Hospitality Management* 89: 102546.