Group 39: MGT 6203 Group Project Final Report

Authors: Josh Bielenberg, Joseph Kim, Nigel Yee, Ying-Kai Huang

Introduction and Background

Cancellation rates pose a significant challenge in the hotel industry, with guests often booking rooms well in advance only to cancel at the last minute. These cancellations result in lost revenue for hotel chains, making it crucial to strike a balance between flexibility for guests and revenue optimization.

Here, we present the project "Hotel Cancellation Prediction." Our goal was to develop a classification model that predicts whether a hotel booking will be canceled or paid for. By leveraging available data and advanced statistical modeling techniques, we aim to identify key factors contributing to booking cancellations and build a reliable predictive model.

Through the analysis of the dataset and the application of suitable modeling approaches, we seek to provide insights into the drivers of booking cancellations and identify the characteristics of bookings most likely to be canceled. The findings from our project can support hotel chains in making data-driven decisions to optimize revenue, strike the right balance between flexibility and revenue protection, and enhance overall customer satisfaction.

This report will outline our methodology, present results, discuss any adjustments made, and provide an overview of the upcoming work. By sharing our progress and analysis, we aim to provide insights that can inform strategies for minimizing cancellations, maximizing revenue, and improving operational efficiency within the hotel industry.

Approach and Methodology

The data set being used includes a dataset of 119390 hotel bookings, and 32 features of each booking. Most notably, is the *is_canceled* feature, which is a binary value that denotes if the booking was ultimately canceled or not. This feature will be used as the response variable in almost all of our modeling. Put differently, we are trying to use all the other data points about a hotel booking to learn about their relationship to if a hotel booking is ultimately canceled.

Because our response variable is a binary value, the natural starting point for modeling is a logistic regression model. Our plan is to start off by fitting a logistic regression model by naively using as many variables as possible in the model. From there, we can get some baseline metrics for naive model performance, and then refine our model from there.

Refinement will happen along two different paths. The first level of refinement is within the features. We hypothesize that additional features may be derived from existing columns that provide additional predictive power. We also anticipate that feature selection and rank reduction will be important, as many columns will be highly correlated with each other.

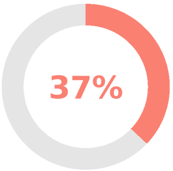
We also intend to refine our model by attempting more advanced modeling techniques such as random forest decision trees. When fitting these models we will be considering if the additional model performance justifies the potential lack of explainability. This tradeoff would ultimately be decided in a business setting where stakeholders would discuss the primary use case of the model. Whether to understand the drivers of cancellations, or to use predictive modeling in some real time use case.

However, before truly tackling any predictive models, data exploration must be done. The 31 other features are a mix of booleans, integers, and categorical variables. In addition, each of these variables will have its own distribution and outliers that must be considered when modeling the data. More discussion about the predictors can be found in the next section.

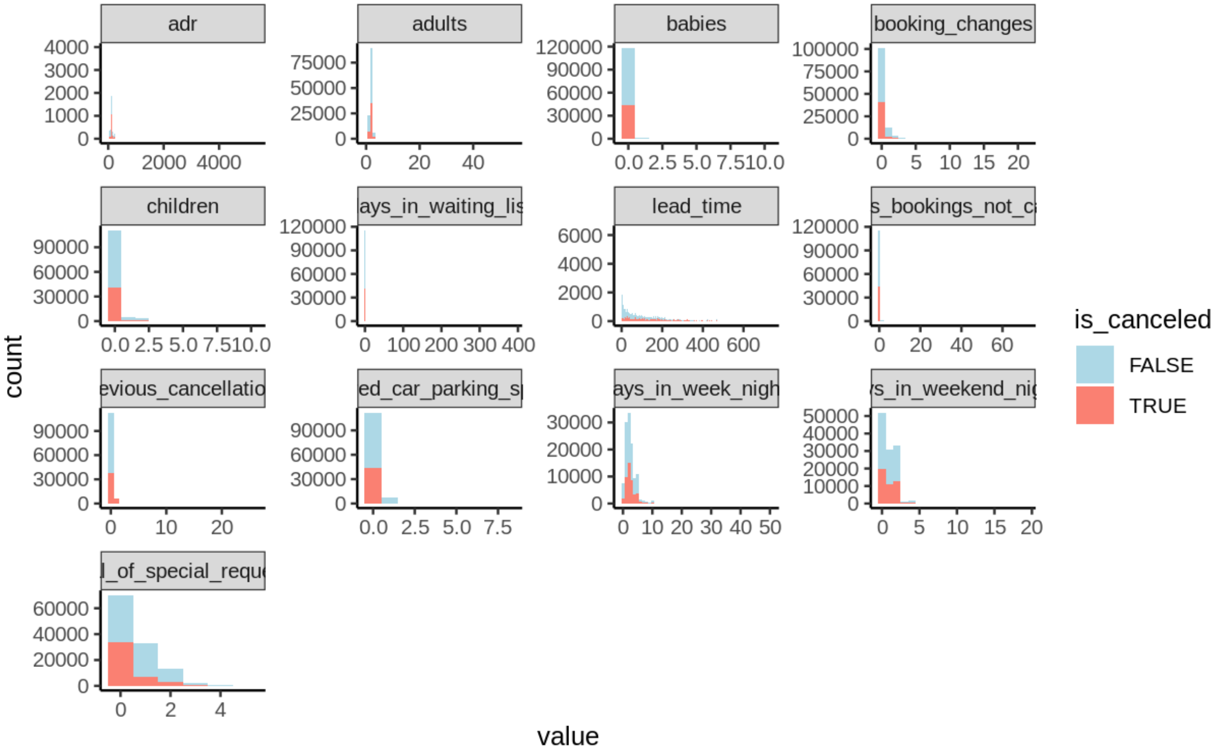
Data Cleaning

Before starting the data cleaning process, EDA must proceed to gain an understanding of what needs to be done.

We first start by assessing the response variable itself, to get an understanding of the likelihood of a cancellation, absent any modeling or predictive features. Overall, **37%** of all bookings are canceled. Which means that if we were to do no modeling with the other 31 features, we would give every booking that came in a roughly one third chance of being canceled.



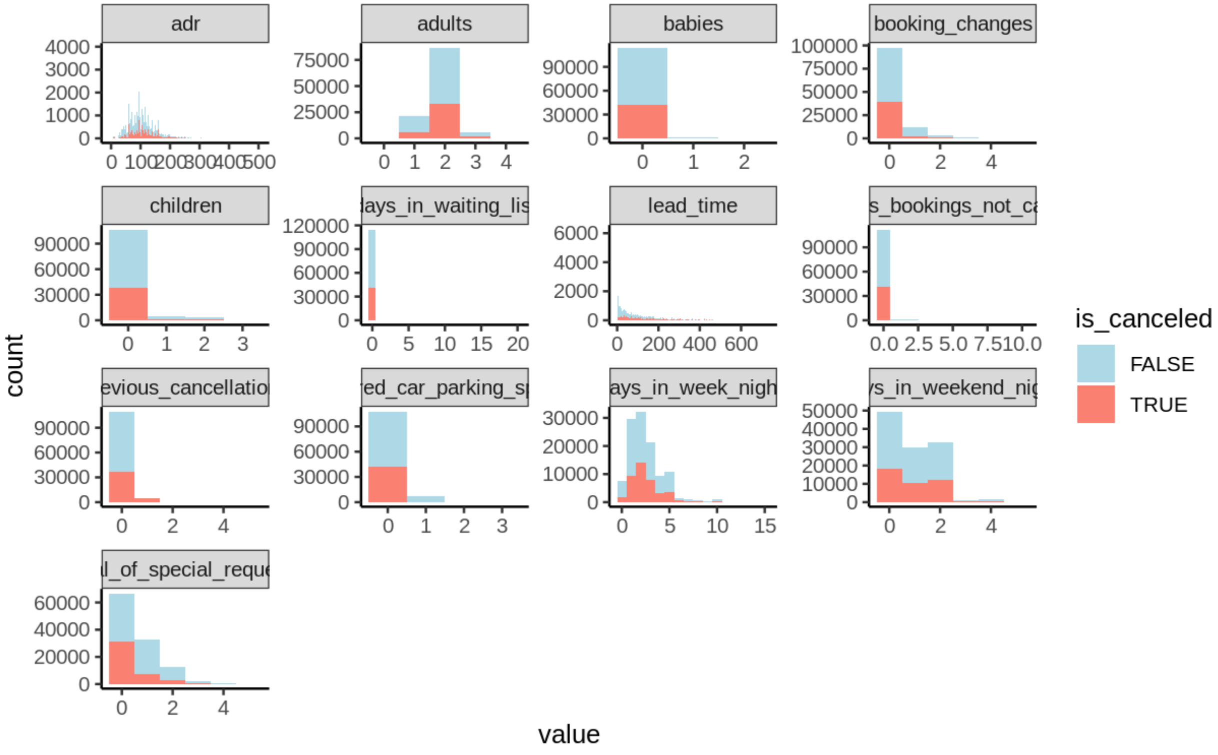
However we believe we can do better than that. To proceed, we need cleaner data to build our models. We first start by assessing the distribution of the quantitative variables. The below chart shows a histogram of values, where the fill line of each chart shows the number of observations with that value for each cancellation status.



Some of these variables have a fairly range of values, like ADR or lead time. Others have a small range of possible values, such as number of babies. This makes sense intuitively as well. However, many of these variables are heavily influenced by extreme outliers. This can be seen by observing the wide x axis with no visible values at the end for variables like ‘booking_changes’ or days_in_waiting list. As an informal measure, let’s set some cutoff values for each of these variables and remove values that don’t make sense. We’ll use the visuals and our intuition for now, but we may revisit with a more formalized method for identifying and removing high leverage points.

We did not use a standard method of removing values if they are more than 1.5 * IQR units away from the 75% percentile because it was too aggressive a measure. This is because so many of these values are heavily concentrated on just one value, like adults.

After removing values based on these thresholds (as a starting point), we are presented with this same visual which shows a distribution of quantitative variables that is less skewed.



Removing these high leverage points will help us get a more realistic model that fits test data better. Additionally, it is reasonable to remove some of these points because this data appears to be influenced by real world errors that impact data quality. For example, a room booking with 40 adults is almost surely an error in booking, which is further supported by the fact that the booking was canceled. It is easy to image bookings being made with a ‘fat fingered’ data input, being canceled, and then rebooked to fix the error.

In regards to the date time variables, we made the choice to remove these because we felt it would be out of scope to add a time series component for a first iteration of this model. Additionally, adding time series predictors would make the interpretability of the model more difficult. However, we anticipate that this could be a fruitful topic for future exploration, in a real world setting where a model would be iterated on quickly, and MVP versions would be put in use to get real world value before seeking additional improvements. However, we did choose to use ‘arrival_date_month’ as a categorical value, to leverage some easily interpretable seasonality effects in our model.

We now move on to examine the categorical variables. First, we had to remove certain categorical variables which we were unable to use, regardless of cleaning. The ‘reservation_status’ was unable to be used because it had a clear relationship with the is_cancelled response variable. We suspect that the is_canceled variable in the data was actually derived from this reservation status column. Therefore, in a real world setting this data point would not be available to predict if a hotel cancellation would be booked *before it was canceled.*

Additionally, we chose to exclude the ‘agent’, ‘company’, and ‘country’ categorical values. These fields had too many distinct data points (330, 331, and 178 respectively) to include as categorical values. If we had, then the data would have been partitioned too finely when modeling and we risked overfitting / overestimating the impact of any individual value of these columns.

Once left with the categorical values we would use, we then examined the relative proportion of rows for each value in the below table.

Predictor	Predictor Value	Number of Rows	Relative Proportion		Predictor	Predictor Value	Number of Rows	Relative Proportion
hotel	City Hotel	75755	66%		deposit_type	No Deposit	102304	89%
hotel	Resort Hotel	39367	34%		deposit_type	Non Refund	12667	11%
arrival_date_month	August	13771	12%		deposit_type	Refundable	151	0%
arrival_date_month	July	12536	11%		distribution_channel	TA/TO	94180	81.81%
arrival_date_month	May	11222	10%		distribution_channel	Direct	14435	12.54%
arrival_date_month	October	10606	9.21%		distribution_channel	Corporate	6314	5.48%
arrival_date_month	April	10587	9.20%		distribution_channel	GDS	192	0.17%
arrival_date_month	June	10482	9.11%		distribution_channel	Undefined	1	0.00%
arrival_date_month	September	9961	8.65%		market_segment	Online TA	56248	48.86%
arrival_date_month	March	9454	8.21%		market_segment	Offline TA/TO	22186	19.27%
arrival_date_month	February	7854	7%		market_segment	Groups	18232	15.84%
arrival_date_month	November	6586	6%		market_segment	Direct	12532	10.89%
arrival_date_month	December	6496	5.64%		market_segment	Corporate	5002	4.34%
arrival_date_month	January	5567	4.84%		market_segment	Complementary	685	0.60%
assigned_room_type	A	70517	61.25%		market_segment	Aviation	237	0.21%
assigned_room_type	D	24973	21.69%		meal	BB	88970	77.28%
assigned_room_type	E	7660	6.65%		meal	HB	13729	11.93%
assigned_room_type	F	3706	3.22%		meal	SC	10590	9.20%

assigned_room_type	G	2522	2.19%	meal	Undefined	1087	0.94%
assigned_room_type	C	2342	2.03%	meal	FB	746	0.65%
assigned_room_type	B	2073	1.80%	reserved_room_type	A	82057	71.28%
assigned_room_type	H	711	0.62%	reserved_room_type	D	19034	16.53%
assigned_room_type	I	345	0.30%	reserved_room_type	E	6428	5.58%
assigned_room_type	K	260	0.23%	reserved_room_type	F	2876	2.50%
assigned_room_type	P	12	0.01%	reserved_room_type	G	2076	1.80%
assigned_room_type	L	1	0.00%	reserved_room_type	B	1103	0.96%
customer_type	Transient	87080	75.60%	reserved_room_type	C	929	0.81%
customer_type	Transient-Party	23434	20.40%	reserved_room_type	H	601	0.52%
customer_type	Contract	4054	3.50%	reserved_room_type	P	12	0.01%
customer_type	Group	554	0.50%	reserved_room_type	L	6	0.01%

Once examining the categorical values, we felt like most groups had sufficient data points to not overfit for that category, with few exceptions. We decided to remove the data points where the reserved, or assigned room type was one of ‘P’ or ‘L’. And we also removed data points where the distribution channel was ‘undefined’. These values happened at such an infrequent rate that we thought they were likely to not be legitimate data points. And if they were, our model would severely overestimate their effects.

At the end of our data cleaning, we were left with 115103 data points. This represents 96.4% of the original data points, with about 4000 data points being removed. We think the cleaned data will produce a model that is much less likely to be overfit on high leverage points.

Additionally, after examining the categorical values, we had the idea to create a derived feature for modeling, is_reserved_and_assigned_room_same’. This was created as a boolean column which denotes if the reserved and assigned room were the same values.

Correlated Predictors

Before proceeding, we wanted to check one last thing, which was if any of the columns in the remaining data were highly correlated with each other. If this were the case, then we would need to address this by dropping one of the features from the model training. If we did not, we would have an unstable model, where small changes in certain predictors would lead to unpredictably large and disproportionate changes in the output.

To examine this, we fit a naive logistic regression model on the data and used the cars package to check the VIF of each predictor.

Predictor	GVIF	Df	GVIF^(1/(2*Df))
is_repeated_guest	1.42E+00	1	1.193426
lead_time	1.46E+00	1	1.208154
stays_in_weekend_nights	1.29E+00	1	1.137094
stays_in_week_nights	1.44E+00	1	1.198095
adults	1.31E+00	1	1.146497
children	2.08E+00	1	1.441182
babies	1.03E+00	1	1.013874
previous_cancellations	1.42E+00	1	1.191814
previous_bookings_not_canceled	1.57E+00	1	1.25347
booking_changes	1.06E+00	1	1.027931
days_in_waiting_list	1.01E+00	1	1.006472
adr	2.50E+00	1	1.579645
required_car_parking_spaces	1.00E+00	1	1
total_of_special_requests	1.20E+00	1	1.097122
hotel	1.57E+00	1	1.252019
meal	1.77E+00	4	1.074248
market_segment	7.04E+01	6	1.425497
distribution_channel	2.58E+01	3	1.719411
reserved_room_type	1.07E+07	7	3.17857
assigned_room_type	1.14E+07	9	2.466428
deposit_type	1.09E+00	2	1.022793
customer_type	2.20E+00	3	1.140711
arrival_date_month	1.93E+00	11	1.030275
is_reserved_and_assigned_room_same	5.61E+00	1	2.367586

To our surprise, we saw that none of the variables bore removing, when considering the generalized VIF. We saw that none were more than 5 when adjusted for the degrees of freedom. It is worth acknowledging that our derived predictor had a high VIF before adjusting for DF. We proceeded with inclusion with the intention of keeping an eye out for suspicious values, and revisiting if needed.

Results and Discoveries

Logistic Regression

We started our analysis with a simple logistic regression to determine how effectively we can predict hotel cancellations using all variables, after removing outliers from the data. We used an 80/20 train-test split for this model, randomly assigning data points to each segment.

Logistic Regression with Numerical and Categorical Variables

The regression incorporated both numerical variables - "lead_time", "stays_in_weekend_nights", "stays_in_week_nights", "adults", "children", "babies", "previous_cancellations", "previous_bookings_not_canceled", "booking_changes", "days_in_waiting_list", "adr", "required_car_parking_spaces", and "total_of_special_requests" - and categorical variables - "hotel", "meal", "distribution_channel", "reserved_room_type", "assigned_room_type", "deposit_type", "customer_type", and "arrival_date_month". Applying the 80/20 split to divide the dataset into train and test sets, the performance of the model on the test set is as follows:

sensitivity	83.62%
specificity	95.01%
FP_Rate	4.99%
precision	90.52%
accuracy	90.87%

From these metrics, it's clear that the simple logistic regression performed quite well in predicting hotel cancellations. The sensitivity indicates an 83.62% success rate in correctly identifying canceled hotel bookings. For other metrics, the simple logistic regression also demonstrated commendable performance."

Using Ridge and Lasso for Model Selections

After observing that the simple logistic regression performed quite well for the task of classifying hotel cancellations, we decided to explore the potential benefits of using Ridge and Lasso techniques to adjust the model coefficients and potentially perform model selection.

Prior to applying Ridge and Lasso regression, we standardized the numerical variables to have 0 mean and a standard deviation of 1. After standardization, all numerical variables could potentially exert similar impacts on the prediction outcome. We also created one-hot vectors for the categorical variables, allowing us to incorporate them into the Ridge and Lasso logistic regressions.

Ridge Regression

We initially adopted ridge regression. To fit this model, we employed the same approach to divide the dataset into training and test sets. We also used cross-validation within the training set to fine-tune the lambda hyperparameter, selecting the one with the best performance on the validation set. The performance of the ridge regression is as follows:

sensitivity	78.7%
specificity	95.7%
FP_Rate	4.3%
precision	91.2%
accuracy	89.5%

By using ridge, we further improve the specificity, precision and false positive rate but sacrifice sensitivity and overall accuracy.

Lasso Regression

We used the same procedure to train and tune the lasso regression model. We ended up getting better sensitivity, specificity and accuracy but worse false positive rate and precision.

sensitivity	89.29%
specificity	93.52%
FP_Rate	6.48%
precision	88.71%
accuracy	91.99%

Although we weren't able to improve performance across all five metrics, it's notable that lasso and ridge affect different sets of metrics. This suggests a potential benefit in combining both methods using an elastic net, which could possibly yield a model that enhances all five metrics simultaneously.

Using Elastic Net for Model Selections

We further investigated the possibility of improving the logistic regression model using elastic net. So far, we have observed some improvement in our model using lasso and ridge techniques, but not across all five metrics utilized for model performance evaluation. By employing the elastic net, we sought to harness the benefits of both lasso and ridge methods to enhance all evaluation metrics.

During the elastic net training, we needed to tune two hyperparameters. Alpha, the parameter governing the balance between ridge and lasso, and lambda, the parameter penalizing larger coefficients in the regression. We divided the range between 0 and 1 into ten sections with an increment of 0.1, selecting alphas and lambdas based on the root-mean-square error during cross-validation.

sensitivity	89.29%
specificity	93.51%
FP_Rate	6.49%
precision	88.69%
accuracy	91.98%

The best alpha selected by cross-validation was 0.9, implying that most of the weight was placed on the lasso. Therefore, we obtained results similar to those of the lasso but with a slight decline in some metrics. This indicates that finding improvements for the model, even when using the elastic net, is not straightforward.

Random Forest

A random forest model was built to see how well hotel cancellations could be predicted, and to determine which features would be good predictors of hotel cancellation. Random forest is an algorithm that builds multiple decision trees with a different set of features used for each tree. As the features selected for each tree are random, the resulting model generally avoids overfitting to the training data, and as multiple trees are built, the performance tends to be relatively good.

For the random forest model, the cleaned hotel bookings dataset was split into 70% for training the model, and 30% for testing. The hyperparameters that need to be tuned for random forest are the number of variables at each split (mtry) and the number of trees (ntree). The mtry was tested, with values from 1 through 10, and ntree = 300, and the test and train error rates were as follows:

mtry	Train Error	Test Error
1	25.11	24.72
2	17.86	17.29
3	15.96	15.74
4	15.08	14.8
5	14.42	14.17
6	14.09	13.85
7	13.95	13.81
8	13.9	13.62
9	13.9	13.65
10	13.97	13.8

Based on the training and test errors of the models, the performance increased with mtry, until mtry=6, where the errors tended to stabilize. While the performance looked to taper off, mtry=8 did have the lowest test error, so this was the selected mtry value for the final model.

The initial model was built using ntree=300. Different ntree values were tested to see the effect that it would have on the resulting model.

mtry	ntree	Train Error	Test Error
8	300	13.9	13.62
8	500	13.89	13.7
8	1000	13.85	13.66
8	1500	13.85	13.69

It was unexpected that increasing the number of trees would decrease the test error, as it was assumed that increasing the number of trees would result in a better trained model. It can be noted that the training error decreased slightly with increasing ntree, however the best test error was found for the lowest ntree value. However, as the difference between the performance of each of these models was negligible, additional testing is needed to come to a definitive conclusion.

Given that ntree=300 resulted in the best test error, this would be the value for the final random forest model.

With the hyperparameters of the random forest model selected, the resulting model gave the following classification results for predicting hotel cancellations:

Training data confusion matrix

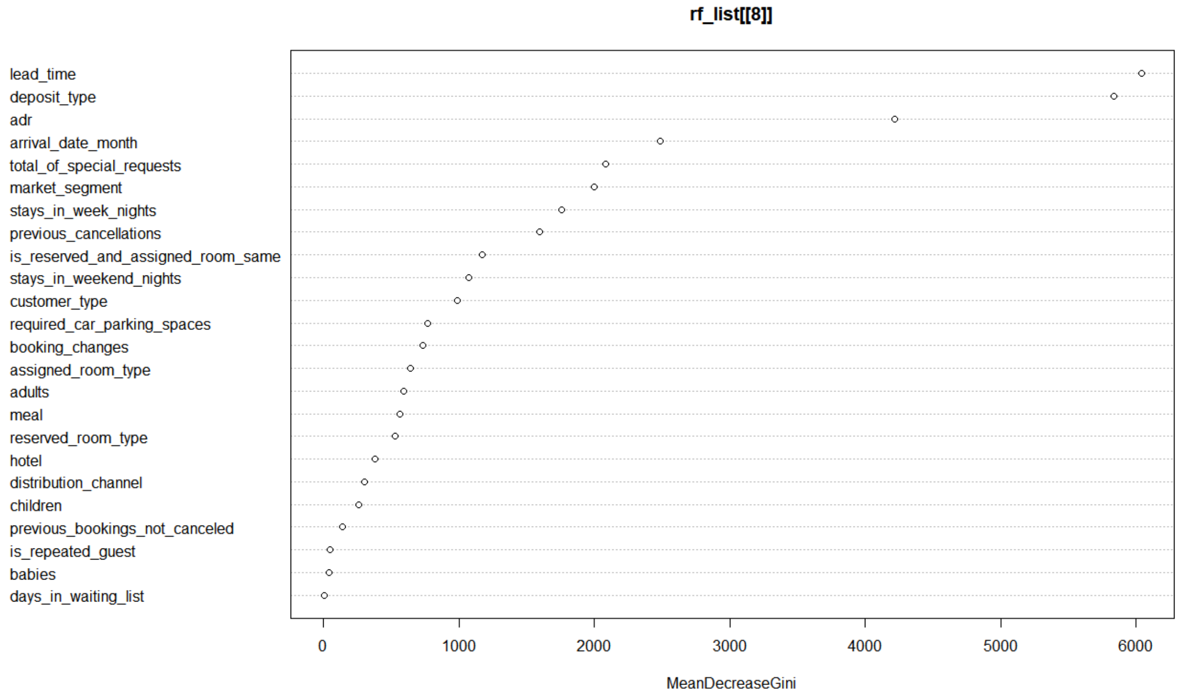
		Actual Values		Class Error
		0	1	
Predicted Values	0	47048	4294	0.0836
	1	6907	22324	0.2363

Testing data confusion matrix

		Actual Values		Class Error
		0	1	
Predicted Values	0	20166	1837	0.0835
	1	2867	9660	0.2289

It can be noted that the model was better at predicting when the hotel reservations were not canceled as opposed to when the hotel is canceled. A possible explanation for this would be that there is more data for when reservations are not canceled as opposed to when they are canceled, so the model was better trained for those scenarios.

Random forest models can also show which features were the most important for the model's performance. The final random forest model provided the following importance plot:



The 3 most important features as determined by the model were ‘lead_time’ (number of days booking was made before the arrival date), ‘deposit_type’ (no deposit, non-refundable, or refundable), and ‘adr’ (average daily rate). The deposit type intuitively makes sense as a good predictor for whether a hotel booking will be canceled as non-refundable bookings are less likely to be canceled as opposed to refundable bookings or bookings without deposits. Likewise with lead time, it’s less likely that someone who books a last minute hotel reservation will cancel as opposed to a booking that was made in advance where plans can change. The average daily rate was a little less expected, however it could stand to reason that if someone booked a more expensive hotel, they are less likely to cancel as a cancellation would be more costly.

Model Comparisons

Comparing the performance of the logistic regression and the random forest models, it was found that the logistic regression model had better performance overall compared to the random forest model.

	Logistic Regression	**Random Forest**
Sensitivity	83.62%	77.11%
Specificity	95.01%	91.65%

Accuracy	90.87%	86.38%

The logistic regression model was better at all metrics, and notably better at predicting whether the booking would be canceled (sensitivity). While both models provided insights to understanding the factors that affect hotel cancellations, it would be recommended to use the logistic regression model that was built to better predict cancellations.

Literature Survey

1. **‘Predicting hotel booking cancellations to decrease uncertainty and increase revenue’, Antonio, N., de Almeida, A. & Nunes, L. (2017)**

In their 2017 study, Antonio, de Almeida, and Nunes used information from four hotels in Portugal's Algarve to create models for forecasting cancellations of reservations. The study's main goal was to show the financial benefits of precisely forecasting cancellations in the hospitality sector.

To examine the dataset, the study used a variety of methods, including tree-based models. The results showed that the tree-based models had the best accuracy, with a about 95% accuracy rate, when averaged over the four hotels. The validation set technique was used to carry out the validation procedure.

The study also showed that there were differences in the prediction models' performance between the hotels. Across all hotels, no particular model consistently outperformed the others. The important factors also varied among the hotels. For instance, the variable "land of origin" had a comparatively less impact on cancellations in the other two hotels while having a more substantial impact in two hotels.

Overall, the results highlighted how critical precise cancellation prediction is to the hotel sector. The study showed that using tree-based models can produce accurate findings that are positive. To maximize revenue management and lessen the financial effect of cancellations, hotel operators should take into account the variations in model performance and important factors among various establishments.

2. **‘Prediction of Hotel Booking Cancellation using CRISP-DM’, Andriawan et al. (2020)**

Andriawan, Z. A., Purnama, S. R., Darmawan, A. S., Wibowo, A., Sugiharto, A., Wijayanto, F. et al. conducted a study in 2020 titled "Prediction of Hotel Booking Cancellation Using CRISP-DM." In order to maximize hotel reservations and reduce income loss, the research sought to investigate the use of machine learning approaches in forecasting hotel booking cancellations. To efficiently analyze and understand the data, the study used the CRISP-DM (Cross-Industry Standard Process for Data Mining) technique.

The study's data came from two separate sources: a vacation hotel in Portugal's Algarve and a downtown hotel in Lisbon. Both datasets had the same 31 variables that represented distinct hotel bookings and the same basic structure. The study applied 10-fold cross-validation using four distinct tree-based models to evaluate their performance. Random Forest outperformed the other models, with an accuracy rate of 87%.

Lead-time, or the number of days between the date of reservation and the date of arrival, was shown to have the greatest impact on the Random Forest algorithm. This result highlights the need of taking lead-time into account when forecasting hotel cancellations.

The study's conclusion is that machine learning can effectively forecast booking cancellations, limiting revenue loss for hotels. Hotel operators may improve their revenue management methods and make wise decisions to reduce the financial effect of cancellations by utilizing the insights learned from the study.

In conclusion, Andriawan et al. (2020) showed the value of using machine learning techniques, in particular the CRISP-DM approach, to forecast cancellations of hotel reservations. The results of the study highlight the relevance of lead-time as a significant factor in the prediction process. Hotels may improve their revenue management techniques, lower income loss, and streamline their booking procedures by utilizing the power of machine learning.

Applications

1. Logistic Regression

For each reservation, the logistic regression model may offer a probability score that indicates the possibility of cancellation. Hotels can use this likelihood to decide how to distribute resources, prioritize bookings, and make other decisions. For instance, they can concentrate on fulfilling reservations with high cancellation probabilities or provide incentives to clients who are likely to cancel. As an early warning system, incorporate the logistic regression model into the hotel's reservation system. The model can instantly determine the likelihood of a cancellation for each new booking, enabling the hotel to take preventative action as needed. By segmenting clients based on their propensity to cancel, hotels can customize their services and marketing approaches for various customer groups.

2. Random Forest

We can use a random forest model to determine the factors that have the greatest impact on booking cancellations. Utilizing this data, hotels can focus their marketing efforts on the elements that result in more confirmed reservations. By properly anticipating the likelihood of cancellations, random forests can aid in the optimization of overbooking methods. Hotels can utilize this data to calculate the ideal level of overbookings, lowering the possibility of having empty rooms as a result of cancellations. To create dynamic pricing techniques, we can combine demand information with the random forest forecasts. In order to increase revenue and occupancy, hotels can modify pricing in real-time by taking the likelihood of cancellations into account.

3. Ensemble Model

Compared to individual models, ensemble approaches frequently offer forecasts that are more reliable and precise. To evaluate the risk connected to a given booking period or event, we can use an ensemble model. Based on the overall risk of cancellations over various time periods, hotels can manage their operations and marketing activities. By utilizing an ensemble prediction-based client retention strategy, hotels can actively engage with customers who are more likely to cancel their reservations, offer them tailored incentives, or offer great customer service to keep their bookings.

Potential Future Work

There are a number of avenues that could be pursued if one wanted to pick up where this project leaves off, to gain additional insights into hotel cancellation rates.

There is an entire time series aspect of the data modeling that was excluded for the sake of scoping. However this could be explored to identify the trend of cancellations over time, and how that trend impacts the likelihood of a cancellation.

For more accurate predictions on cancellation rates, ensemble models and Synthetic Minority Over-sampling Technique (SMOTE) can be used to balance the classes. Ensemble models can be created by combining the predictions from multiple models, such as logistic regression, random forest, and other classifiers like support vector machines or gradient boosting.

Given available data regarding customer reviews on their stay experiences, we can incorporate sentiment analysis on such reviews related to the hotel. Sentiment analysis can help identify potential issues that may lead to cancellations and give insights into areas where improvements can be made.

A deeper analysis could be made into the interpretation of these models, to create more robust conclusions about the drivers behind each predictor's impact on the model.

Finally, this hotel cancellation model could be paired with a pricing strategy analysis, to figure out the optimal amount of overbooking that a hotel may employ. This would balance the tradeoffs in the reduced losses from unfilled rooms with the increased losses due to paying out overbooked guests who are unable to stay in their room.

Conclusion

Despite the challenges involved with fitting a predictive classification model on real world data, with all its issues included, we believe the models created for this project could bring value to companies all across the hospitality industry. We were able to create multiple models with ~90% accuracy, and were able to show that the most important factors to consider when trying to predict if a hotel will be canceled is the lead time, deposit type, and average daily rate of the room.

References

Andriawan, Z. A., Purnama, S. R., Darmawan, A. S., Wibowo, A., Sugiharto, A., Wijayanto, F. et al. (2020), Prediction of hotel booking cancellation using crisp-dm, in ‘2020 4th International Conference on Informatics and Computational Sciences (ICICoS)’, IEEE, pp. 1–6.

Antonio, N., de Almeida, A. & Nunes, L. (2017), ‘Predicting hotel booking cancellations to decrease uncertainty and increase revenue’, Tourism & Management Studies 13(2), 25– 39.