Springboard – DSC

Analysis and Predictive Modeling of Hotel Cancellation and Price Trends

Final Report

Soyoung An

Dipanjan Sarkar (Mentor)

TABLE OF CONTENTS

1 Introduction

Hotels provides services for people on holiday that deals with guest accommodation or lodgings. However, hotel cancellation is one of the biggest challenges to control in hospitality industry. Hotel cancellation can lead to the business loss. Having ability to accurately predict future customers' cancellation rates can help the business gain and future expected revenue. In addition, when we can use cancellation prediction to forecast the potential cancellation rate of a particular customer, it allows to target the individuals to prevent them from cancellation.

Predicting a hotel price rate is essential to evaluate demand for accommodation and pricing the room rates accordingly. Hotel industry is an area where there is a lot of competitions. Having lower rate with same or advance accommodation will help marketing and acquiring customers. Therefore, predicting the actual demand and price can help to reduce the unexpected profit loss. This can help the industry by reducing the unexpected risk, as well as ready with enough facilities.

1.1 Objectives

The objectives of this project are to:

- Explore and analyze hotel booking data
  - Identify the key factors that influence cancellation rate and price
  - Identify whether the average daily rates (ADR) are same between two groups (cancelled and not cancelled)
- Develop a predictive model to estimate the cancellation
- Develop a model to predict hotel price considering multiple factors

1.2 Significance

By analyzing the historical hotel related data, we will identify the main factors that influence hotel churns and price. Any hospitality industry business such as hotels.com, bookings.com, Airbnb, or hotel booking agencies can use suggested predictive models to check their price and cancellation rates.

2 Dataset

2.1 Data Description

The dataset used in this study are sourced from the website Kaggle, a subsidiary of Google LLC, is an online community of data scientist and machine learning practitioners that allows users to find and publish datasets, explore and build models in a web-based data-science environment. This consist of a dataset: hotel_bookings.csv file:

- Hotel_booking.csv file is composed of total of 32 columns of features and 40,060 observations of H1 (resort hotel) and 79,330 observations of H2 (city hotel) collected from the article Hotel Booking Demand Dataset written by Nuno Antonio.
- H1 is a resort hotel located in Algarve, Portugal and H2 is a city hotel located in Lisbon, Portugal. Both datasets comprehend bookings due to arrive between the July 1, 2015 and August 31, 2017, including bookings that effectively arrived and bookings that were cancelled.

A description of each of the features of the dataset are provided in Table 2.1.

Table 2.1 Description of hotel_bookings.csv

No	Feature Name	Feature Description	Data Type
1	hotel	Resort Hotel (H1) or City Hotel (H2)	Object
2	is_canceled	Value indicating if the booking was cancelled (1) or not (0)	Integer
3	lead_time	Number of days between the booking date to the arrival date	Integer
4	arrival_date_year	Year of arrival	Integer
5	arrival_date_month	Month of arrival	Object
6	arrival_date_week_number	Week number according to year of arrival	Integer
7	arrival_date_day_of_month	Day of arrival	Integer
8	stays_in_weekend_nights	Number of weekend nights booked (Saturday and Sunday)	Integer
9	stays_in_week_nights	Number of weeknights booked (Monday to Friday)	Integer
10	adults	Number of adults	Integer
11	children	Number of children	Integer
12	babies	Number of babies	Integer
13	meal	Type of meal booked	Object
14	country	Country of customer's origin	Object
15	market_segment	Market segment designation	Object
16	distribution_channel	Booking distribution channel (how the booking was made)	Object
17	is_repeated_guest	Value indication if the booking was from a repeated guest (1) or not (0)	Integer
18	previous_cancellations	Number of previous cancellations prior to current booking	Integer
19	previous_bookings_not_canceled	Number of previous booking not cancelled prior to current booking	Integer
20	reserved_room_type	Reserved room type code	Object

No	Feature Name	Feature Description	Data Type
21	assigned_room_type	Code for the type of room assigned to the booking	Object
22	booking_changes	Number of changes made to the booking since entering the hotel management system	Integer
23	deposit_type	Type of deposit made for the reservation (No Deposit, Refundable, Non refund)	Object
24	agent	ID of the travel agency that made the booking	Float
25	company	ID of the company/ organization that made the booking or is responsible for payment	Float
26	days_in_waiting_list	Number of days booking was in the waiting list until it was confirmed	Integer
27	customer_type	Type of booking	Object
28	adr	Average Daily Rate (the sum of transactions divided by the number of nights stayed)	Float
29	required_car_parking_spaces	Number of car parking spaces requested	Integer
30	total_of_special_requests	Number of special requests made by the customer	Integer
31	reservation_status	Last reservation status (Cancelled, Check-Out, No-Show)	Object
32	reservation_status_date	Date at which the last status was set	Object

3 Package Introduction

In this study we used Jupyter Notebook to run all the codes for data analyzing and modeling. Numpy, Pandas, Matplotlib, Seaborn were installed as basic package. Datetime was installed to manipulate dates and times. Statistical functions (scipy.stats) were imported for inference statistical test. Scikit-learn was installed as the machine learning library. Tabulate is installed to print tabular data as formatted tables.

4 Data Wrangling

4.1 Dataset Processing

We processed the dates type data by following 6 steps. More details of each step are included in Table 4.1.

Table 4.1 Data Processing Steps for Dates

Steps	Action	Variable Names	Detail Explanation
Step 1	Data type Correcting	reservation_status_date	Convert object to datetime
Step 2	New Variable Creation	arrival_date	Add new variable ‘arrival_date’ in datetime by combining ‘arrival_date_year’, ‘arrival_date_month’, and ‘arrival_date_day of month’
Step 3	Data Transformation	arrival_date_month	Convert month names to numeric value
Step 4	New Variable Creation	reservation_status_date_year	Add new variable ‘reservation_status_date_year’ by extracting year value from variable ‘reservation_status_date’
		reservation_status_date_month	Add new variable ‘reservation_status_date_month’ by extracting month value from variable ‘reservation_status_date’
		reservation_status_date_day	Add new variable ‘reservation_status_date_day’ by extracting day value from variable ‘reservation_status_date’
		reservation_status_day_of_week	Add new variable ‘reservation_status_day_of_week’ by extracting day name from variable ‘reservation_status_date’
		arrival_date_day_of_week	Add new variable ‘arrival_date_date_day_of_week’ by extracting day name from variable ‘arrival_date’
Step 5	Rename	arrival_date_day	Rename ‘arrival_date_day_of_month’ to ‘arrival_date_day’
Step 6	Removal	reservation_status_date	At the end of data wrangling, reservation_status_date was removed. All information (year, month, day) is stored separately.
		arrival_date	At the end of data wrangling, removed arrival_date. All information (year, month, day) is stored in separately.

Missing Values

We found some missing values in children, country, agent, and company columns in the dataset.

- ‘company’ column was dropped since it has 94% of columns with missing values.
- Missing values in ‘agent’ column were replaced with a value zero, representing ‘not a third party’.
- Missing values in ‘country’ column was replaced with a string ‘not available’.
- There are 4 missing values in children and filled with 0 since the majority children values are zero.

Suspicious Data

- Average Daily Rate (adr)

We investigate the average daily rate with zero, negative, and very small values. We were not able to find any additional information such as rewards program (free-night) or policy. Histogram below is shown in Figure 4.1 that represent the count numbers for average daily rate under 30 euro. We dropped the 0 adr for the histogram for the scaling reason. We were able to see a big drop at 25 euro adr and decided to investigate further more.

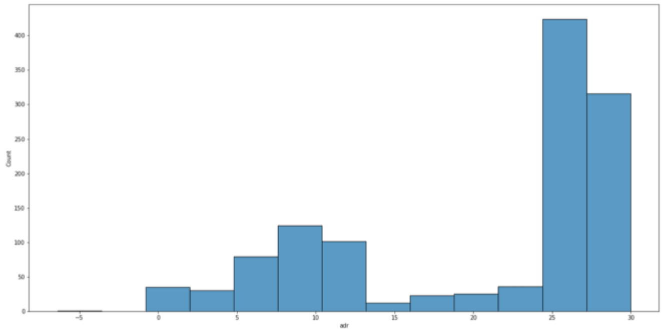


Figure 4.1 Histogram of average daily rate below 30 euro (dropped 0 adr for scaling)

There are 2437 rows containing adr values less than 25 euro. The size of data volume (adr less than 25 euro) is minor compare to the total size of the dataset. We decided to drop these rows. More detailed information is shown in Table 4.2.

Table 4.2 Number of rows in different conditions

Condition	Number of Rows	adr values
adr < 0	1	-6.38
adr = 0	1959	0
0 < adr < 25	477	0.26 to 24.95
Total	2437	-6.38 to 24.95

Also, there was one outlier row with adr value of 5400 euro. We dropped this row for data analysis and modeling.

- No-Show

There was one row with no-show in reservation with different reservation_status_date and arrival_date. No-Show should be considered as not showing on the day of arrival date and the data was ambiguous. We decided to drop this one row.

- Final Shape of Dataset

After all the dataset processing, there are 116,951 rows and 35 columns left from 119,390 rows and 32 columns initially.

5. Exploratory Data Analysis

5.1 Summary Statistics

After cleaning the data, statistics including mean, standard deviation, minimum, maximum, and percentiles for each variable in the hotel_booking.csv were summarized in Table 5.1. Excluded classification data types.

Table 5.1 Summary Statistics

Vaiable Name	Count	Mean	Std	Min	25%	50%	75%	Max
lead_time	116951	105.188130	106.924572	0	19	71	162	709
stays_in_weekend_nights	116951	0.935358	0.993030	0	0	1	2	19
stays_in_week_nights	116951	2.519517	1.885362	0	1	2	3	50
adults	116951	1.861831	0.480757	0	2	2	2	4
children	116951	0.104223	0.398829	0	0	0	0	10
babies	116951	0.007867	0.097193	0	0	0	0	10
is_repeated_guest	116951	0.027730	0.164198	0	0	0	0	1
previous_cancellations	116951	0.081906	0.777167	0	0	0	0	26
previous_bookings_not_canceled	116951	0.125258	1.448456	0	0	0	0	72
booking_changes	116951	0.215193	0.630094	0	0	0	0	18
days_in_waiting_list	116951	2.345367	17.710354	0	0	0	0	391
adr	116951	103.863971	46.422812	25	71.1	95	126	510
required_car_parking_spaces	116951	0.062607	0.245517	0	0	0	0	8
total_of_special_requests	116951	0.571718	0.791880	0	0	0	1	5

The average leading time between arrival date and booking date is 105 days which is about 3 months and more. The average number of adults is 2 people. The mean of previous cancellations variable is 0.08 however, there are some extreme cases where customers made up to 26 times previous cancellations. The average time waited in the waiting list until it is confirmed is a little bit over 2 days. The maximum waited days in waiting list is 391 days. The mean of average daily rate is 103.86 euro with 46.42 standard deviation. The lowest price is 25 euro and highest is at 510 euro.

5.2 Heatmap and Correlations

Heatmap is shown in Figure 5.1 to see the relationship and correlation amongst the variables.

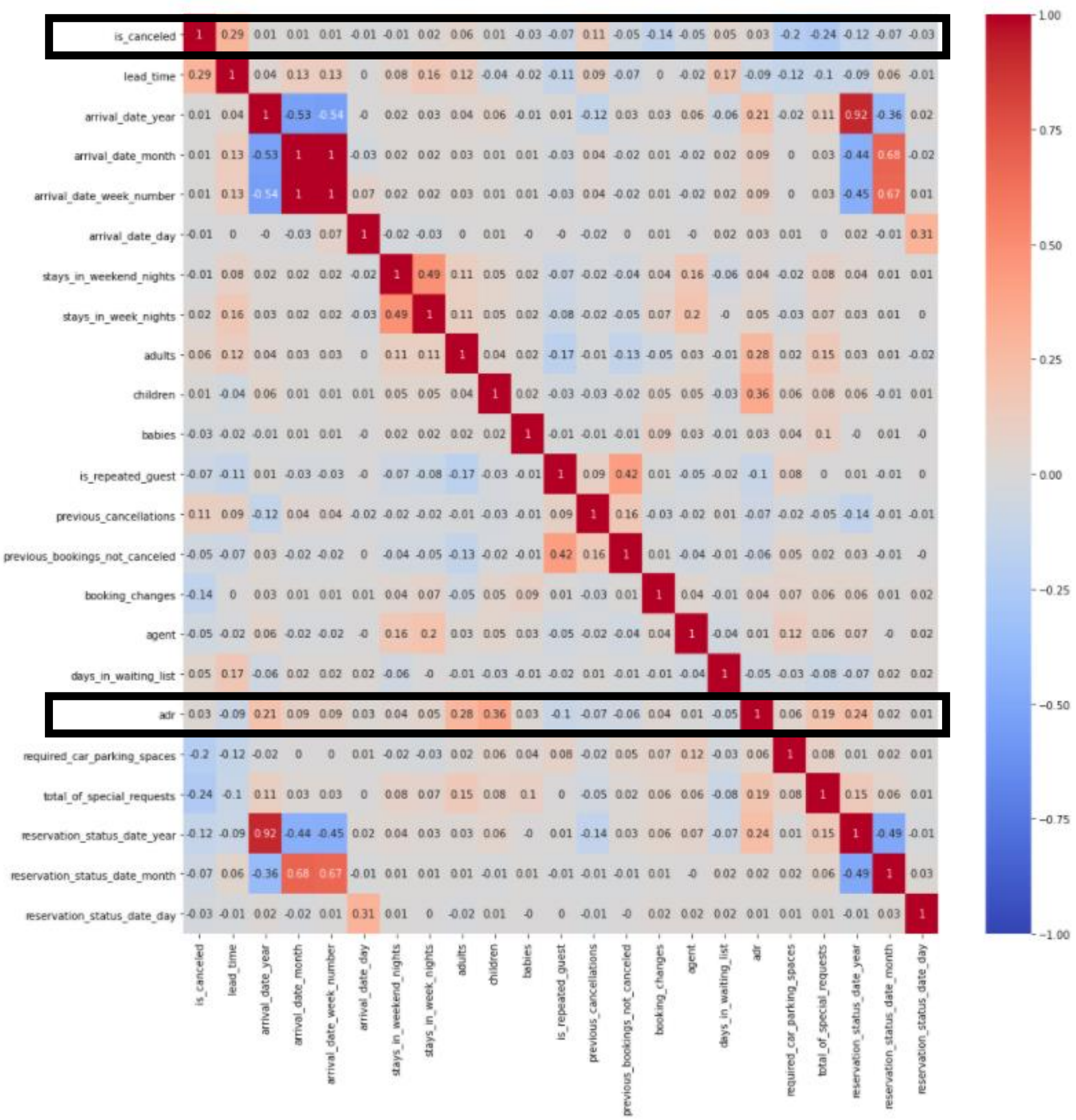


Figure 5.1 Heatmap of correlation coefficients between variables

Average daily rate (adr) has high correlation with children, adults, reservation_status_date_year, arrival_date_year, and total_of_special_requests. Having more people (adults and children) and special orders increases daily rate.

Cancellation (is_canceled) has high association with lead_time, previous_cancellations, days_in_waiting_list, and adr. Having longer waiting time until the booking date and long waiting time until the booking confirmation often cause cancellations. Also, customer’s history and price are important features that affects cancellation.

5.3 Dataset Explorations

There are 73,140 reservations that were checked-in (not cancelled), and 43,811 reservations that were cancelled. From Figure 5.2, you can find the count plot for cancellation.

Also, there were similar numbers of reservation bookings throughout the week for both for cancelled and non-cancelled groups. In Figure 5.2, we visualized the number of reservations by different day of week and cancellation.

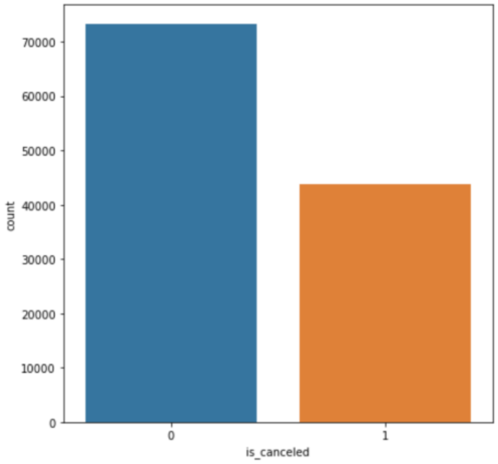


Figure 5.2
Count plot of different cancellation types

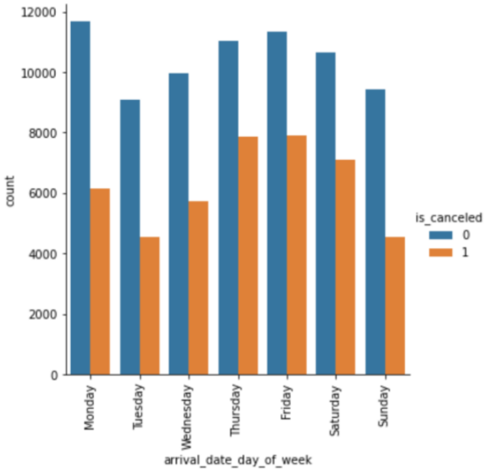


Figure 5.3 Count plot by day of week of booking reservations and cancellation

5.4 Hypothesis Test and Bootstrapping

We performed hypothesis test to see whether if the average daily rate price booked by who has cancelled has the same or different rates as people who is not cancelled.

- Null hypothesis: The average daily rate price (adr) booked by who has cancelled the reservation has the same rate as people who is not cancelled.
- Alternative hypothesis: The average daily rate price (adr) booked by who has cancelled the reservation does NOT have the same rate as people who is not cancelled.

We performed the hypothesis test using t-test. The shape of histogram follows normal distribution trends however, it is slightly skewed to the right (Figure 5.4). Therefore, we used bootstrapping method to check the t-test results.

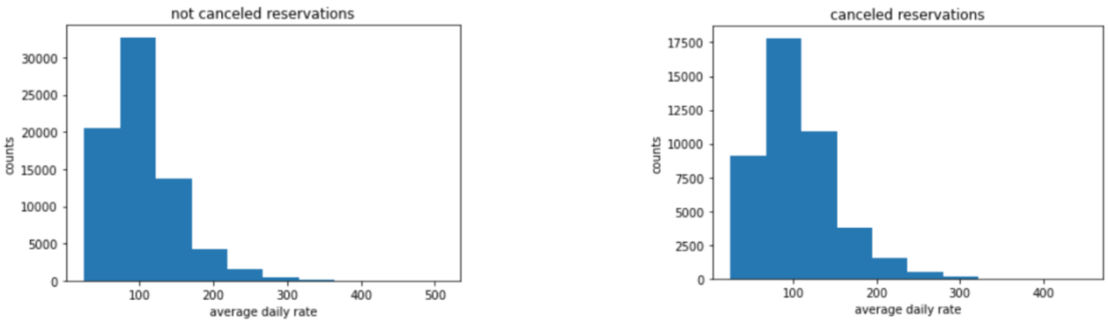


Figure 5.4 Histogram of average daily rate by different cancellation types

Both t-test using original dataset and bootstrapped sample, we obtained p-value lower than 0.05, standard alpha value which also known as significance level. Therefore, we can safely reject the null hypothesis and accept alternative hypothesis which the average daily rate (adr) between two groups (who has cancelled and not cancelled the reservations) are different.

6 Hotel Cancellation Predictive Modeling

Any addition information about code and more details could be found in the jupyter notebook 03(1)_hotel_cancellation_preprocessing_modeling.

6.1 Preprocessing and Train/Test Datasets

Before model training, we took 2 preprocessing steps for hotel cancellation predictive modeling. The details are listed in the Table 6.1.

Table 6.1 Preprocessing steps for Hotel Cancellation Predictive Modeling

No.	Action	Variable Names	Details
1	Removal	reservation_status	The values of this categorical data are ‘Check-Out’ or ‘Canceled’ or ‘No-Show’, identical to target variable, lead to overfitting.
2	Dummy Encode Variable	'hotel','arrival_date_day_of_week','meal', 'country', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type', 'reservation_status_day_of_week'	Creating dummy variables for categorical variables. This step has increased the number of variables from 34 to 260.

Train/Test Split

We dropped the variable ‘is_canceled’ and trained our models with the rest of 259 features. The data was split into 70%/30% training/testing sets and specified on the target variable as ‘is_canceled’ variable.

6.2 Modeling Approach

In this section, we studied the performance of 6 classification models: Logistic Regression, k-Nearest Neighbors, Support Vector Machine (Linear Kernal), Decision Tree, Gradient Boosting, and Random Forest. The pros and cons of each models are summarized in Table 6.2.

Table 6.2 Overview of classifier

No.	Binary Classifier	Advantages	Disadvantages
1	Logistic Regression	• Easy to interpret • Small number of hyperparameters • Overfitting can be addressed though regularization	• May overfit when provided with large numbers of features • Can only learn linear hypothesis functions • Input data might need scaling • May not handle irrelevant features well
2	k-Nearest Neighbors	• No training involved, easy to implement • Only one hyperparameter	• Need to find optimal number of K • Slow to predict • Outlier sensitivity
3	Support Vector Machine	• Accuracy • Works well on smaller cleaner datasets • Is effective when number of dimension is greater than the number of samples	• Is not suited to larger datasets as the training time can be high • Less effective on noisier datasets with overlapping classes
4	Decision Tree	• Easy to interpret • Work with numerical and categorical features • Requires little data processing • Performs well on large datasets • Doesn’t require normalization	• Overfitting • Unable to predict continuous values • Doesn’t work well with lots of features and complex large dataset

5	Gradient Boosting	• Excellent predictive accuracy • Can optimize on different loss functions • Provides several hyperparameter tuning options that make the function fit very flexible	• Overfitting risk • Computationally expensive • Parameter complexity
6	Random Forest	• Excellent predictive power • Requires little data preprocessing • Doesn't require normalization • Suitable for large dataset • Plenty of optimization options	• Overfitting risk • Parameter complexity • Limited with regression

We trained all the model with the baseline implementation, meaning all the hyperparameters of the models were left as the default value in the scikit-learn APIs. The evaluations are all conducted over the same Training set (70%) with 5 Fold Cross Validation. The average performance on the test folds for each model can be found from Table 6.3. (Top 2 best scores for metrics accuracy are highlighted in yellow)

Table 6.3 Average baseline classification models performance on 5 test folds

No.	Model	Cancellation	Accuracy	Precision	Recall	f1_score
1	Logistic Regression (LR)	0	0.71	0.71	0.91	0.8
		1		0.72	0.38	0.5
2	k-Nearest Neighbor (KNN)	0	0.87	0.86	0.94	0.90
		1		0.89	0.74	0.81
3	Support Vector Machine (SVM)	0	0.83	0.84	0.91	0.87
		1		0.82	0.71	0.76
4	Decision Tree (DT)	0	0.95	0.96	0.96	0.96
		1		0.94	0.93	0.94
5	Gradient Boosting (GB)	0	0.90	0.89	0.96	0.92
		1		0.92	0.81	0.86
6	Random Forest (RF)	0	0.95	0.94	0.99	0.96
		1		0.98	0.89	0.93

Most of the models have good accuracy scores (0.95 as the highest, 0.71 as the lowest). Decision Tree and Random Forest models are scored highest with 0.95 accuracy. We have obtained precision, recall and f1 scores respective to cancellation classes where 0 for non-cancelled reservations and 1 for cancelled reservations. Since we are classifying for who is going to cancel the reservation, we will focus on cancellation group 1. Decision Tree model performs slightly better in predicting false positive (precision) than false negative (recall). Random forest showed some difference in false positive and false negative. False negative prediction is larger than false positive which resulted lower recall number to 0.89.

6.3 Tuning Hyperparameters & Imbalance Data

For the further tuning process, we decided to tune hyperparameters of top 2 best performing models: Decision Tree and Random Forest Models.

Decision Tree Model

We first used the RandomizedSearchCV method with 5 cross validations and 200 samples in the conditions as below:

- Criterion: gini or entropy
- Max_depth: from 70 to 199
- Min_samples_split: from 1 to 9
- Min_samples_leaf: from 1 to 9

Using the results from RandomizedSearchCV, we narrowed down the parameter criteria for GridSearchCV as below:

- Criterion: gini or entropy
- Max_depth: from 140 to 175
- Min_samples_split: from 3 to 8
- Min_samples_leaf: from 1 to 5

The best parameter conditions for Decision Tree are listed as below:

- Criterion: ‘entropy’
- Max_depth: 140
- Min_samples_split: 3
- Min_samples_leaf: 1

Random Forest Model

We refined the model using GridSearchCV with hyperparameters as below:

- Max_depth: 100 or 200 or None
- Max_features: ‘sqrt’ or ‘log2’
- N_estimators: 100 or 500 or 1000

The best parameter conditions for Random Forest are listed as below:

- Max_depth: 100
- Max_features: ‘sqrt’
- N_estimators: 1000

Dataset Imbalance

The data imbalance in training/testing set is existed in different cancellation classes. We performed to see if the outcome improves by weighing differently. The performance of both Decision Tree and Random Forest models are same or slightly lower with different ration to handle data imbalance. We were not able find any improvement in the performances of either of models.

6.4 Best Hotel Cancellation Model Selection

The classification report for Decision Tree Model with tuned hyperparameter conditions (criterion='entropy', max_depth=140, min_samples_split=3, min_samples_leaf=1) and Random Forest Model with best hyperparameter conditions (max_depth=100, max_features='sqrt', n_estimators=1000) are resulted in Table 6.4 (Best model is highlighted in yellow).

Table 6.4 Model Performance with tuned hyperparameters for Hotel Cancellation

No.	Model	Cancellation	Accuracy	Precision	Recall	f1_score
1	Tuned Decision Tree (DT)	0	0.96	0.96	0.97	0.96
		1		0.95	0.94	0.94
2	Tuned Random Forest (RF)	0	0.95	0.94	0.99	0.96
		1		0.98	0.89	0.94

The Best Hotel Cancellation Model is the Decision Tree Model with tuned hyperparameters.

6.5 Comparative Studies (Shapley Additive Explanations)

SHAP (Shapley Additive Explanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

We created SHAP Explainer and computed SHAP values for the Hotel Cancellation Model. Due to computing power, we weren't able to compute SHAP values for regression (Hotel Price Model).

Figure 6.1 shows SHAP summary plot for cancellation booking, ranked from highest to lowest association.

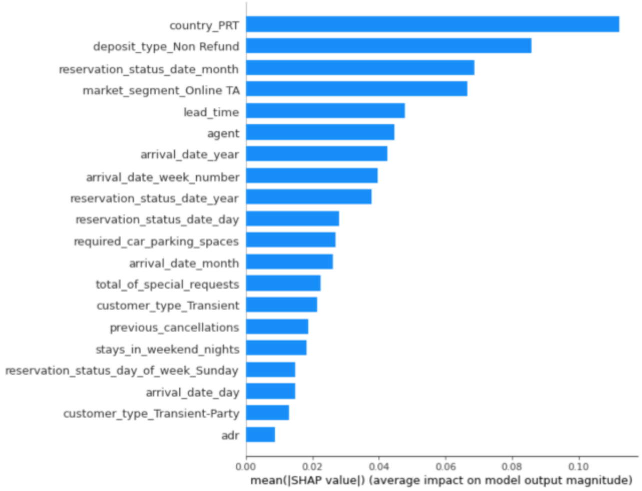


Figure 6.1 SHAP Summary plot for cancelled booking

Most influential features affecting the model prediction of a specific transaction of booking is shown in force plot below. Figure 6.2 is an example of cancelled hotel reservation SHAP force plot. Figure 6.3 represents a SHAP force plot for non-cancelled hotel reservation.

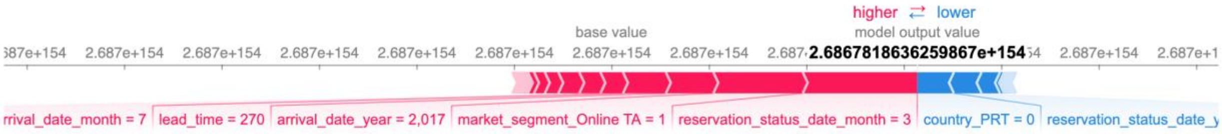


Figure 6.2 Example of Cancelled Reservation SHAP Force Plot

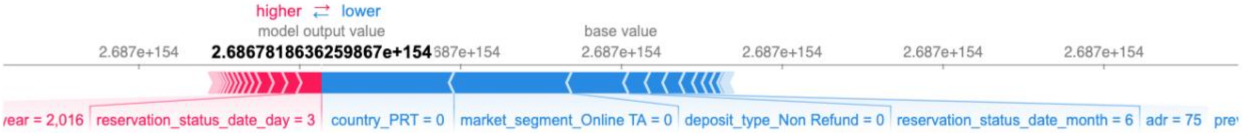


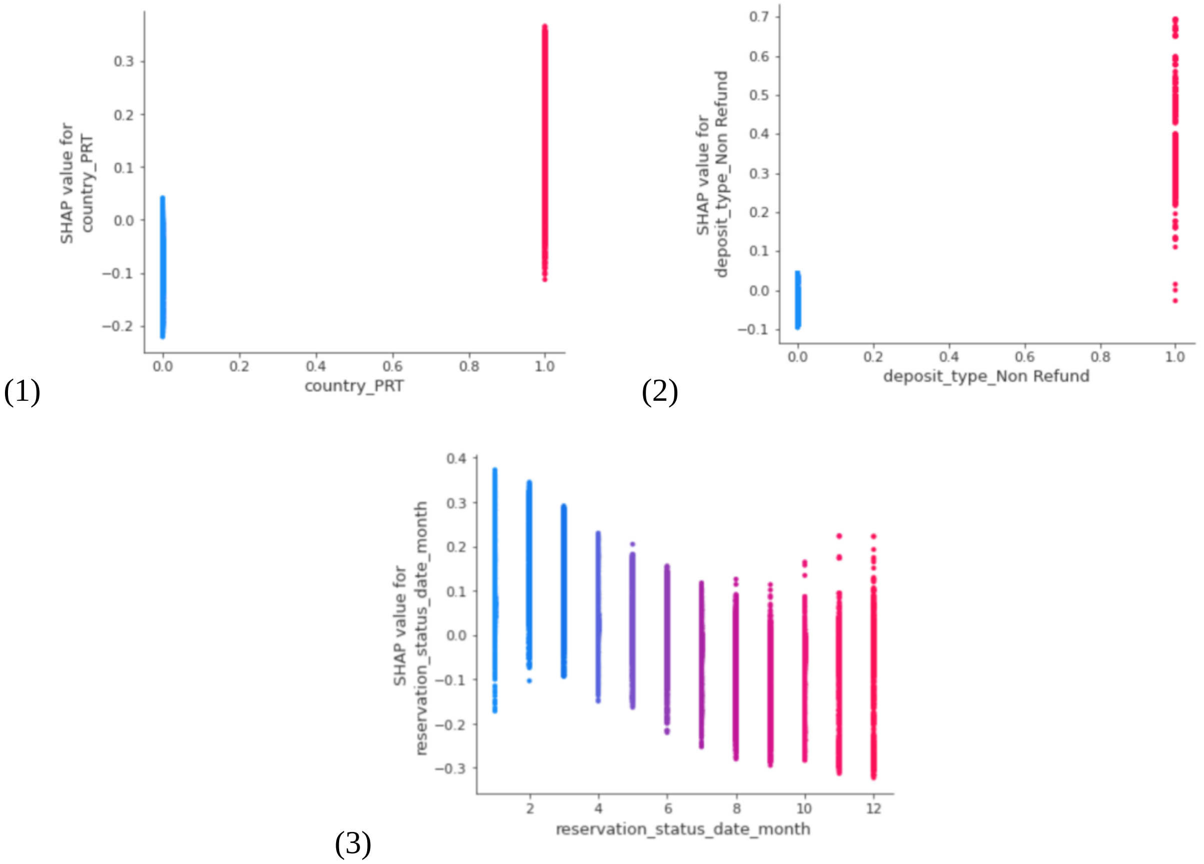
Figure 6.3 Example of Not Cancelled Reservation SHAP Force Plot

Figure 6.4 (1) shows that there is a higher chance that reservation is leading to a cancellation when country code is equal to Portugal. The datasets are obtained from a city and resort hotel in Portugal. There may be more possibilities that people living in same country tend to cancel more easily than different country travelers.

Non-refundable deposit type tends to lead to a cancellation. We were not able to find additional information on the dataset. It would be nice to compare their payment history to see whether their deposit was fully charged. Customers may have booked the reservation but canceled before deposit is fully charged.

Lastly, we investigated that more cancellations occur during the holiday seasons (November – March).

The partial dependence plots for top 3 variables influencing the hotel cancellation model are shown in Figure 6.4.



(1)

(2)

(3)

Figure 6.4 Top 3 variables influencing the hotel cancellation model

7 Hotel Price Predictive Modeling

Any addition information about code and more details could be found in the jupyter notebook 03(2)_hotel_price_preprocessing_modeling.

7.1 Preprocessing and Train/Test Datasets

Before model training, we took 3 preprocessing steps for hotel cancellation predictive modeling. The details are listed in the Table 7.1.

Table 7.1 Preprocessing steps for Hotel Price Predictive Modeling

No.	Action	Variable Names	Details
1	Removal	reservation_status	The values of this categorical data are ‘Check-Out’ or ‘Canceled’ or ‘No-Show’, identical to target variable, lead to overfitting.
2	Dummy Encode Variable	'hotel','arrival_date_day_of_week','meal', 'country', 'market_segment', 'distribution_channel', 'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type', 'reservation_status_day_of_week'	Creating dummy variables for categorical variables. This step has increased the number of variables from 34 to 260.
3	Scale Numeric Variable	'lead_time', 'arrival_date_year', 'arrival_date_month', 'arrival_date_week_number', 'arrival_date_day', 'stays_in_weekend_nights', 'stays_in_week_nights', 'adults', 'children', 'babies', 'is_repeated_guest', 'previous_cancellations', 'previous_bookings_not_canceled', 'booking_changes', 'agent', 'days_in_waiting_list', 'required_car_parking_spaces', 'total_of_special_requests', 'reservation_status_date_year', 'reservation_status_date_month', 'reservation_status_date_day'	Standardizing numerical variables into a scale

Train/Test Split

We dropped the variable ‘adr’ and trained our models with the rest of 259 features. The data was split into 70%/30% training/testing sets and specified on the target variable as ‘adr’ variable.

7.2 Modeling Approach

For the Hotel Price Predictive Modeling, we used similar approach except that it is a regression instead of classification. Here are 6 approaches we used for the hotel price predictive modeling.

1. Linear Regression
2. KNN Regression
3. SVM Regression (Linear Kernal)
4. Decision Tree Regression
5. Gradient Boosting Regression
6. Random Forest Regression

We trained all the model with the baseline implementation, meaning all the hyperparameters of the models were left as the default value in the scikit-learn APIs. The evaluations are all conducted over the same Training set (70%) with 5 Fold Cross Validation. We used R squared (R2) score, root mean squared error (RMSE), and mean absolute error (MAE) for each models. The average performance on the test folds for each model can be found from Table 7.2. (Top 2 best scores for metrics accuracy are highlighted in yellow)

Table 7.2 Baseline regression models performance

No.	Model	R Squared Score	RMSE	MAE
1	Linear Regression	-4538919097154.504	99213625.486	1156328.643
2	KNN Regression	0.74275	23.6194	14.2729
3	SVM	0.46846	33.9517	23.2192
4	Decision Tree	0.86178	17.3130	8.08617
5	Gradient Boosting	0.76437	22.6052	16.2039
6	Random Forest	0.92517	12.7386	6.53402

7.3 Tuning Hyperparameters & Imbalance Data

For the further tuning process, we decided to tune hyperparameters of top 2 best performing models: Decision Tree and Random Forest Models.

Decision Tree Model

We used the RandomizedSearchCV method with 5 cross validations and 200 samples in the conditions as below:

- Splitter: best or random
- Max_depth: [None, 0, 10, 20, 30, … , 190, 200, 210]
- Min_samples_split: from 1 to 9
- Min_samples_leaf: from 1 to 9

The best parameter conditions for Decision Tree are listed as below:

- Criterion: ‘best’
- Max_depth: 30
- Min_samples_split: 7
- Min_samples_leaf: 6

Random Forest Model

We refined the model using GridSearchCV with hyperparameters as below:

- Max_depth: [None, 0, 50, 100, 150, 200]
- Max_features: ‘auto’ or ‘sqrt’ or ‘log2’
- N_estimators: 100 or 500 or 1000

The best parameter conditions for Random Forest are listed as below:

- Max_depth: 50
- Max_features: ‘auto’
- N_estimators: 1000

7.4 Best Hotel Price Model Selection

The model performance on tuned hyperparameters for Decision Tree and Random Forest Regression models are represented in Table 7.3.

Table 7.3 Model Performance with tuned hyperparameters for Hotel Price

No.	Model	R Squared Score	RMSE	MAE
1	Tuned Decision Tree	0.8772157	16.31799	8.737637
2	Tuned Random Forest	0.9259167	12.67521	6.474431

The Best Hotel Price Model is the Random Forest Model with tuned hyperparameters.

8 Conclusions

8.1 Summary

We have investigated hotel datasets to provide the analysis to reduce cancellation and set reasonable price points which helps to reduce unexpected profit loss. We developed both classification and regression models using various machine learning algorithms. We performed comparative studies to analyze how features impacts the results of cancellation models for individual transactions of customers.

8.2 Future Scope

Additional Machine Learning techniques can be applied for both cancellation and price model such as XGBoost or LightGBM methods. Due to time constraint, we decide to move on without trying additional machine learning techniques. Also, Gradient Boosting method didn’t perform as great as expected. Both XGBoost and LightGBM are efficient implementation forms of Gradient Boosting.