Predicting Hotel Booking Cancellation

Predictive Analytics Sec 003

Yash Wali, Somesh Yadav.

04.24.2022

ABSTRACT

We analyze the data related to Hotel booking cancellations to derive significant contributors and Predict cancellations based on various aspects. Predicting hotel booking cancellations is a significant real-world problem and can help organizations and businesses to improve their sales. We implemented several models with varying results, the best model giving us an accuracy of over 90%. We will discuss more about our model comparison later, looking at it from a more of a business perspective.

METHODOLOGY

Our group decided to use Python for this project. The benefits of Python include consistency, flexibility, access to powerful AI and machine learning (ML) libraries and frameworks, platform independence, and large communities.

We begin by extracting the data and analyzing it to get a deeper grasp. The first step is always to check the data quality and perform pre-processing steps to ensure that our data is ready for analysis. Some exploratory data analysis is performed to understand the trends in data. Additionally, it can help us get a sounder understanding of the relationship between different features.

After all these steps, we are ready to implement machine learning algorithms to this data to predict our target variable, "is_cancelled."

Since what we are trying to achieve is, in essence, a binary classification, ROC/AUC score is another critical metric we choose to consider, in addition to the accuracy of our model.

Lastly, results, insights, as well as some shortcomings are identified.

DATA EXPLORATION AND VISUALIZATIONS

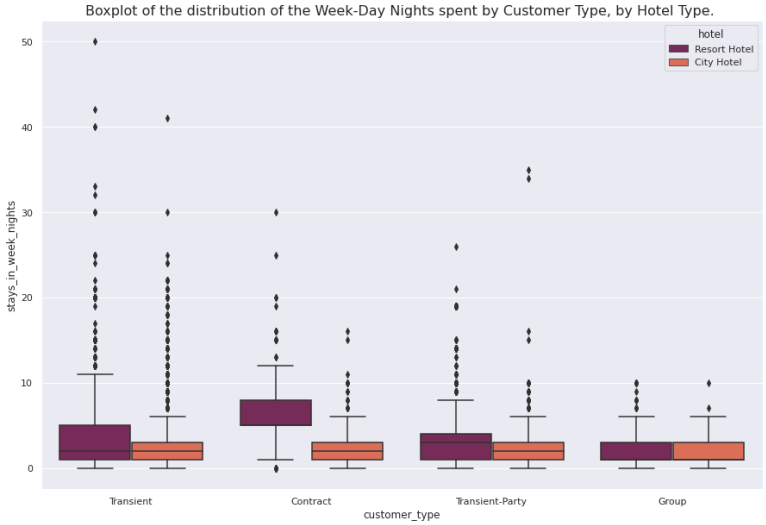
Data Overview - Important Columns

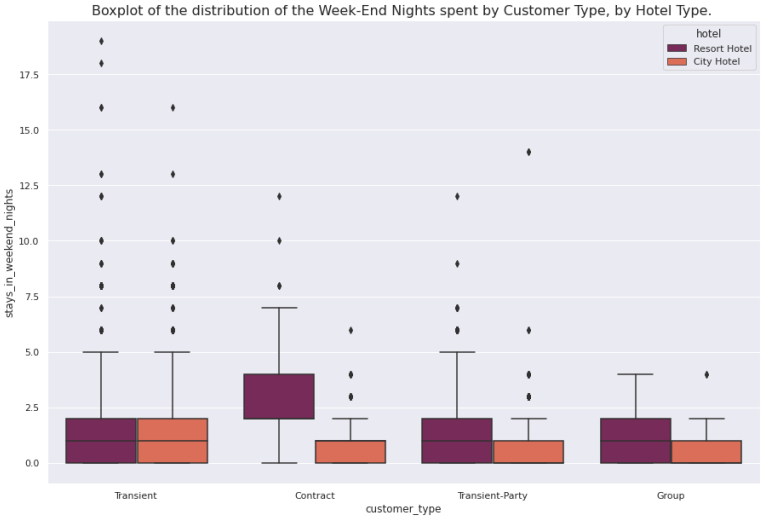
Column Name	Description
is_cancelled	Value indicating if the booking was canceled (1) or not (0).
customer_type	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it; Group – when the booking is associated to a group; 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.
market_segment	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”.
deposit_type	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit – no deposit was made; Non Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of stay
is_repeated_guest	Value indicating if the booking name was from a repeated guest (1) or not (0).
reservation_status	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
lead_time	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date.
adr	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights.

Distribution of the Nights spent by Customer Type, by Hotel Type.

In pretty much all cases, people tend to stay for longer in Resort Hotels than in City Hotels.

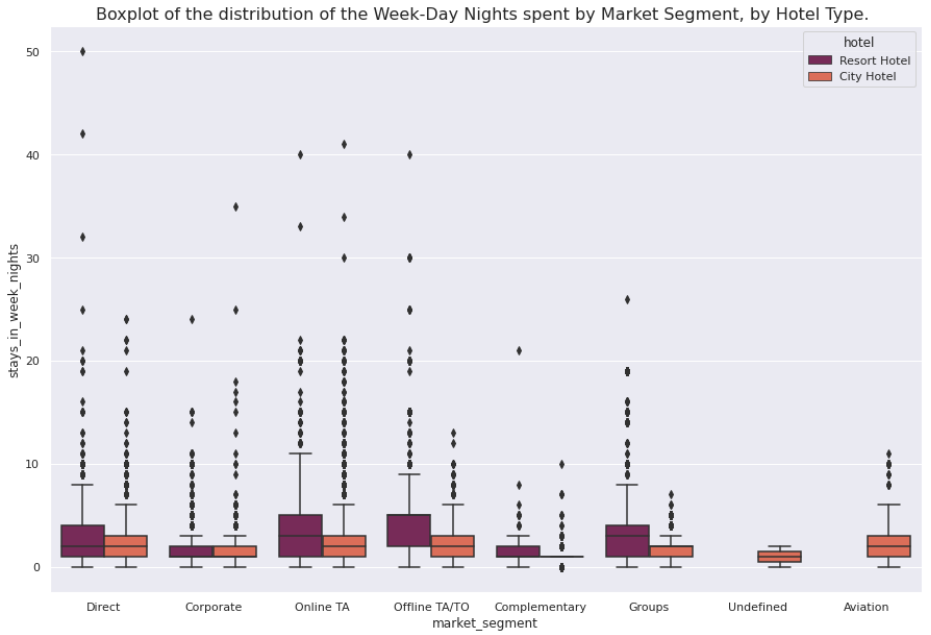
“Contract” type customers who book Resort Hotels stay for the longest periods of time.

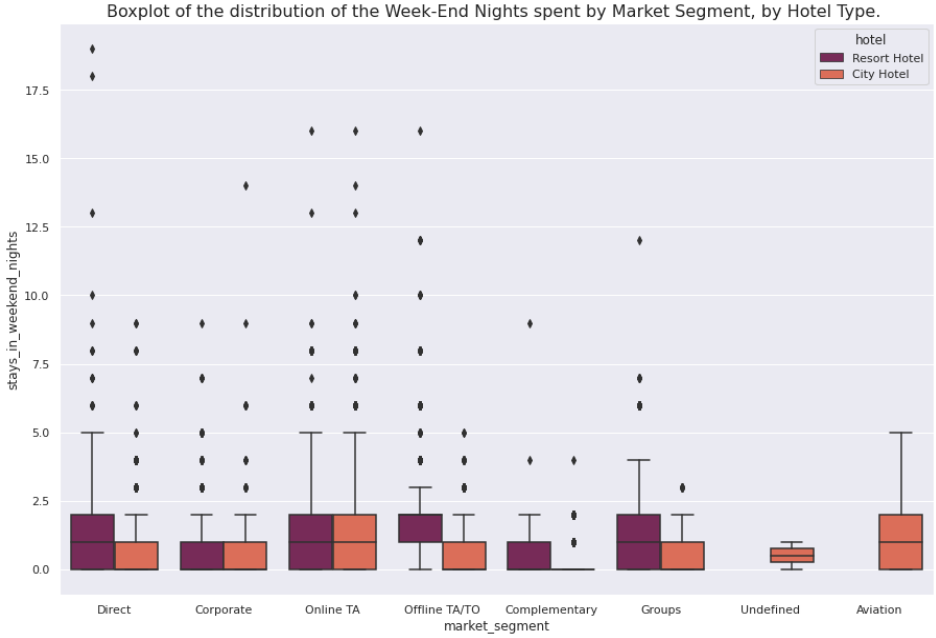




On average, people book hotels for 5 days or less on weekdays, or 2 days or less on weekends implying most people stay in Hotels for a week at most, the sole exception being "Contract" Customers who book Resort Hotels.

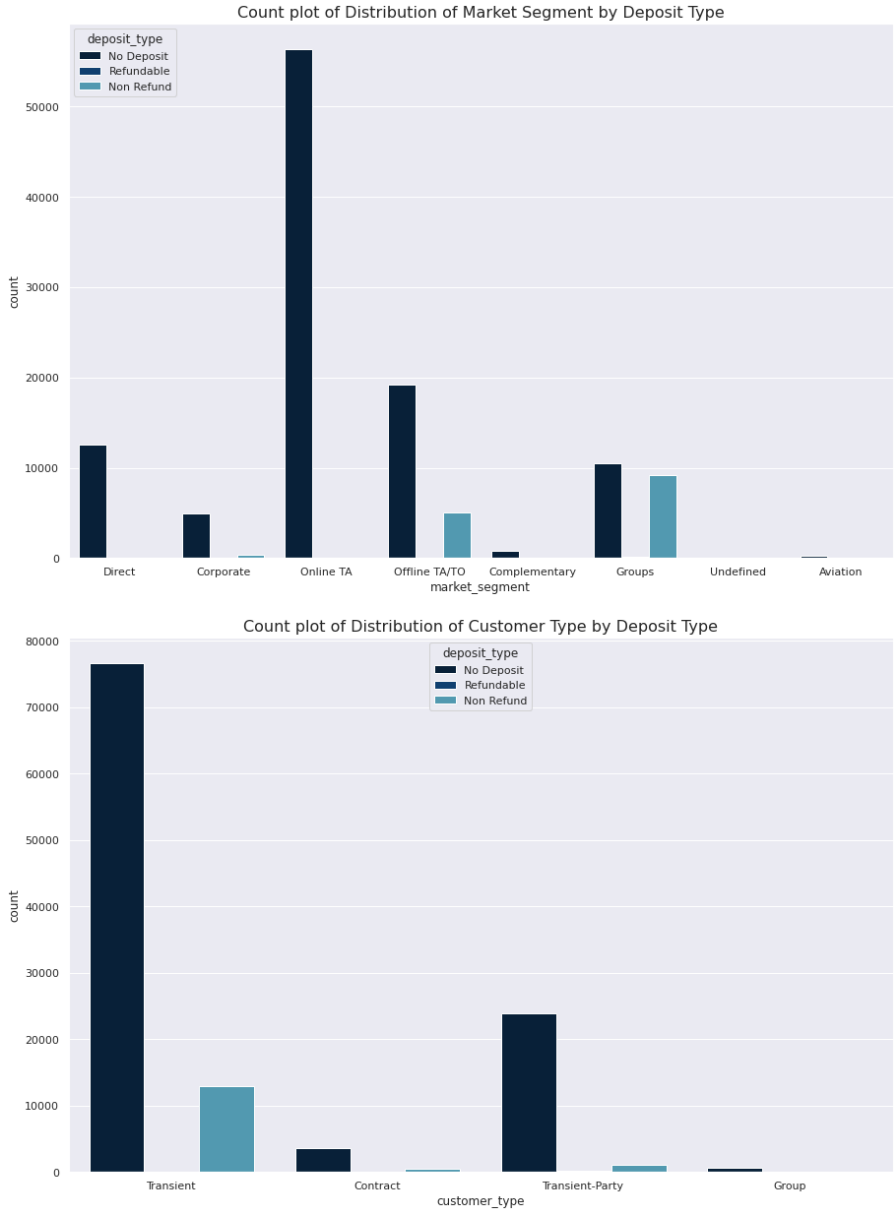
Distribution of the Nights spent by Market Segment, by Hotel Type.





People from the Aviation sector seem to say nearly exclusively in City Hotels. This is related to the fact that most Airports are in or close to Cities.

Distribution of Customer Type and Market Segment by Deposit Type.

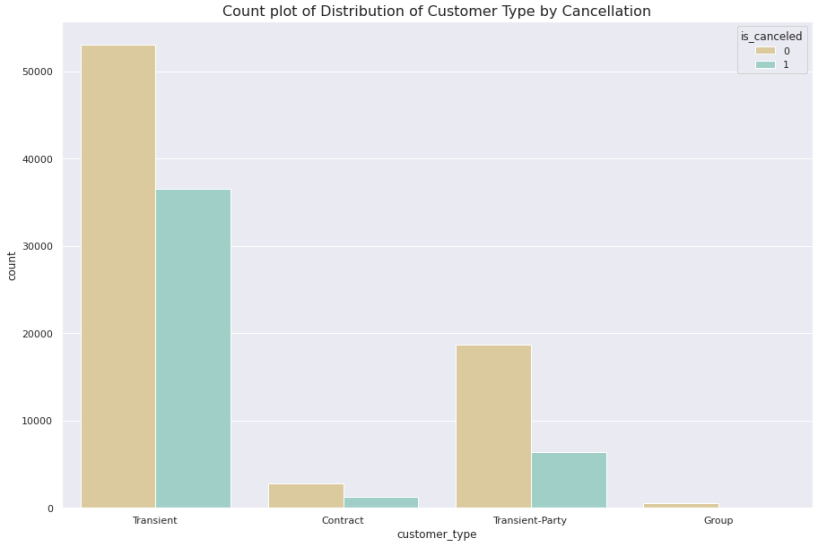


Looking at these plots, most customers, and segments, do not give any deposits while booking (Or choose Hotels where deposits are not required). Only the "Groups" and "Offline TA/TO" Market segment seems to have a relatively higher distribution of Deposits taken.

Now, let's move on to some visualizations related to our target variable, "is_cancelled"

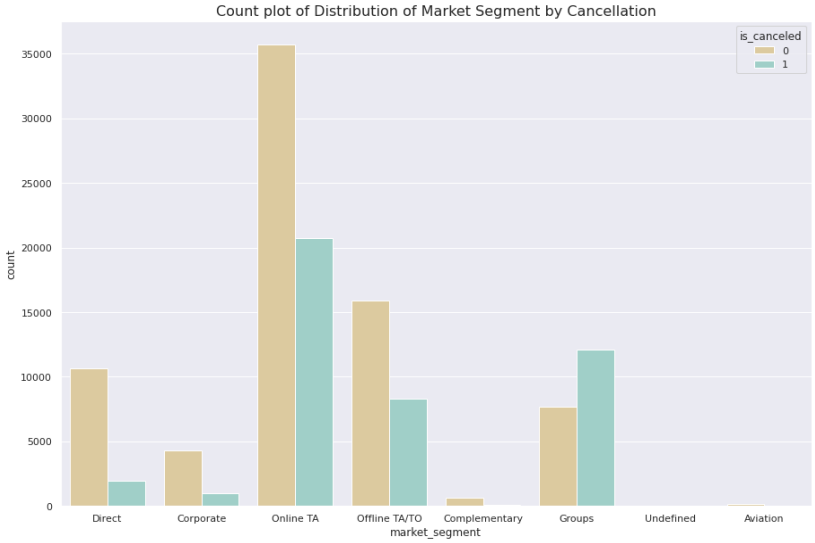
Distribution of Customer Type, and Segment Type, by Cancellation.

Note: **0: Booking Not Cancelled.** **1: Booking Cancelled.**

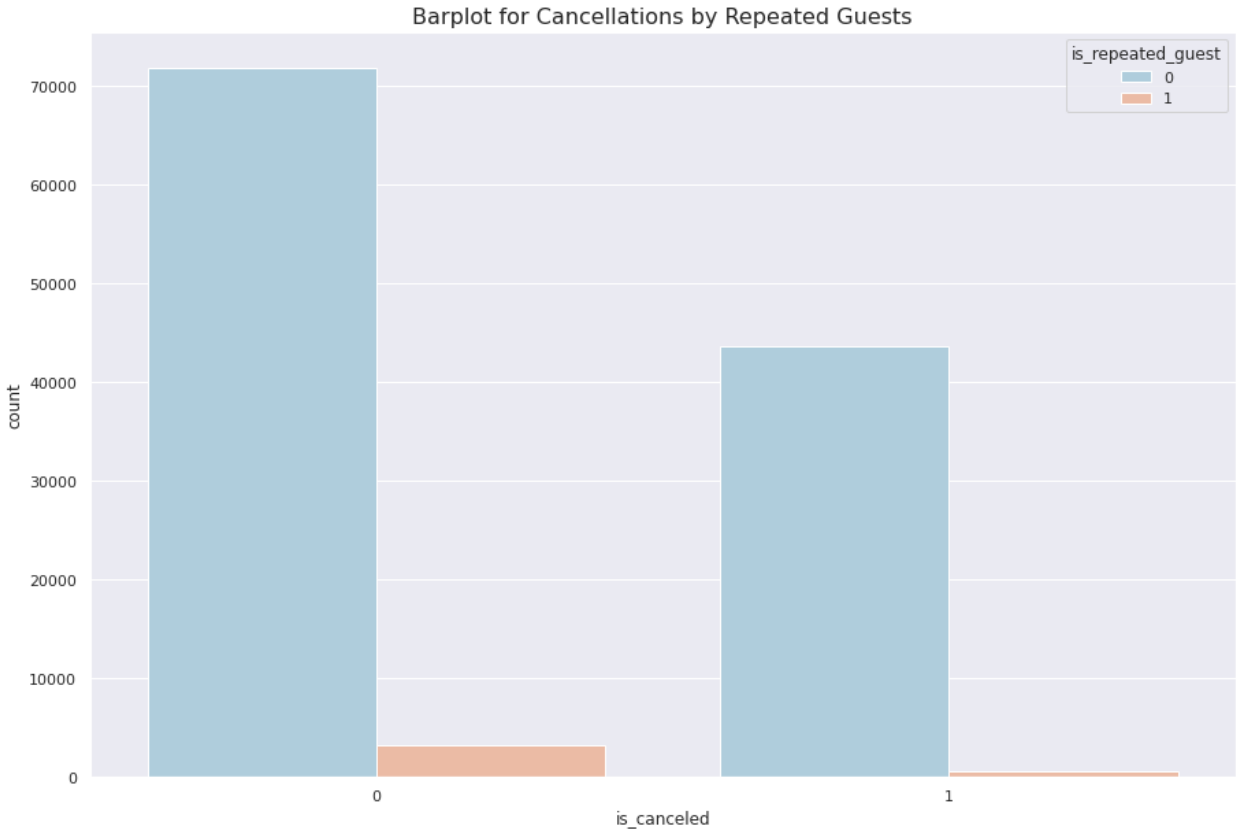


As we can clearly tell, the cancellation rate is very significant. Therefore, the problem we are looking to analyze should be a pretty serious problem for the Hotels, thus our solutions would have significant real-world applications.

From this graph, we can see the Hotel owners would love people who Directly Communicate with them, or with the Corporate Segment of the market, and would especially hate Group Bookings Online. Of course, it seems that people who booked in Groups tend to require a Non-Refundable Deposit more often than other groups, so this is debatable.



Cancellations by Repeated Guests



Proportionately, it is easy to see that repeat guests tend to not cancel as often as new guests. This may be a useful feature for our prediction models later.

DATA PRE-PROCESSING

Null or Missing Values

We take a glance at the missing values in our data. The "agent" and the "company" columns have a high proportion of their values missing. In this case, we opt to drop them.

Additionally, a few rows in the "country" and "children" columns have missing values. Their numbers, however, in comparison to the data we have, are insignificant. We can simply drop those rows.

We decided against using data imputation techniques. Mean imputation, which somewhat makes sense in our case, does not preserve the relationship amongst variables. Furthermore, it leads to an underestimate of standard errors. Moreover, we have enough data that we can simply do away with the missing portion of it at very little or no cost to accuracy.

```
hotel                               0
is_canceled                         0
lead_time                           0
arrival_date_year                   0
arrival_date_month                  0
arrival_date_week_number            0
arrival_date_day_of_month           0
stays_in_weekend_nights             0
stays_in_week_nights                0
adults                              0
children                            4
babies                              0
meal                                0
country                           488
market_segment                      0
distribution_channel                0
is_repeated_guest                   0
previous_cancellations              0
previous_bookings_not_canceled      0
reserved_room_type                  0
assigned_room_type                  0
booking_changes                     0
deposit_type                        0
agent                           16340
company                        112593
days_in_waiting_list                0
customer_type                       0
adr                                 0
required_car_parking_spaces         0
total_of_special_requests           0
reservation_status                  0
reservation_status_date             0
dtype: int64
```

Correlation Analysis

As we can see from the correlation values to the right, "reservation_status" is extremely correlated to our target variable.

```
reservation_status              -0.917223
total_of_special_requests       -0.235643
required_car_parking_spaces     -0.194796
assigned_room_type              -0.175880
```

The reason for this is that "reservation_status" is pretty much the same thing as our target variable, only later in time, and it will not be something we can use for consideration in our model as soon as a customer books a room for obvious reasons.

Therefore, we drop this, and the column "reservation_status_date", an extension of this column.

Feature Type Modification and Undersampling

Lastly, we focus on type modifications for the remaining features. We convert the categorical features to int by one-hot encoding to create the respective dummy features.

Here we face a dilemma. The “country” column is categorical, but with over 150 unique values to it. We can’t possibly create those many dummy variables. This left us with two solutions:

1) Label encode the column, but not on-hot encode. This comes with issues such as a country holding more weight than the other.
2) Drop it altogether.

Ultimately, we decided that we will let the results speak for themselves, and try out both options. Surprisingly, with the label encoded “country” column, we consistently see about a 1-2% increase in accuracy across all models that we tested. Hence, the “country” column narrowly survived its demise.

Finally, we fix the data imbalance issue. Since we have a lot more data for class 0, which corresponds “booking not canceled”, we undersample our data, since we have plenty leftover, to remove this issue.

Now, the data is ready for modeling.

MODELING

Multiple classification algorithms are selected and trained on our given data, and accuracy scores and ROC-AUC curves are used to determine the quality of the model.

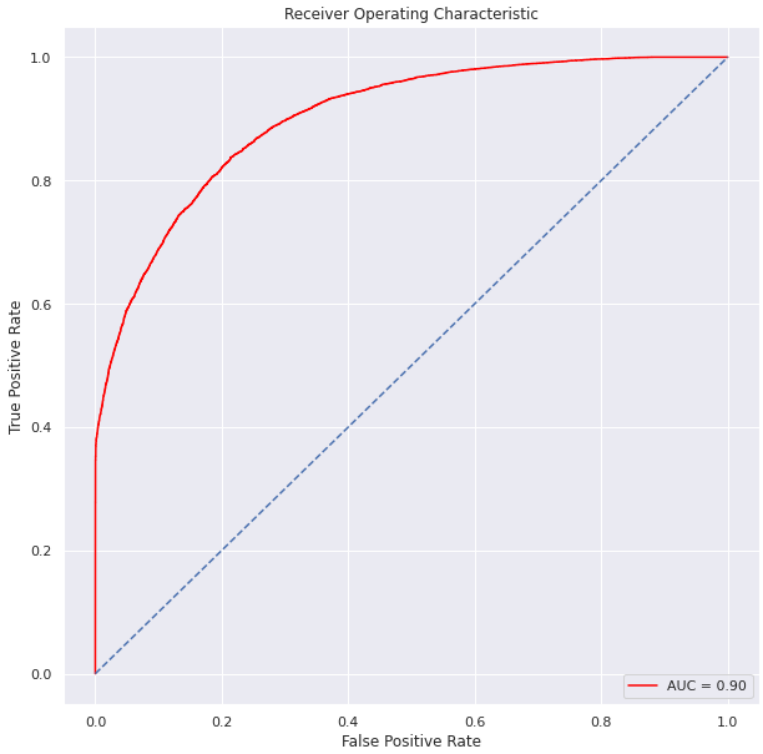
Since this is a binary classification problem ROC-AUC scores are of great importance.

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

Logistic Regression

Since our target variable is binary, Logistic regression is the first algorithm that we implemented for this purpose. Logistic regression has high interpretability in terms of its working and feature selection.

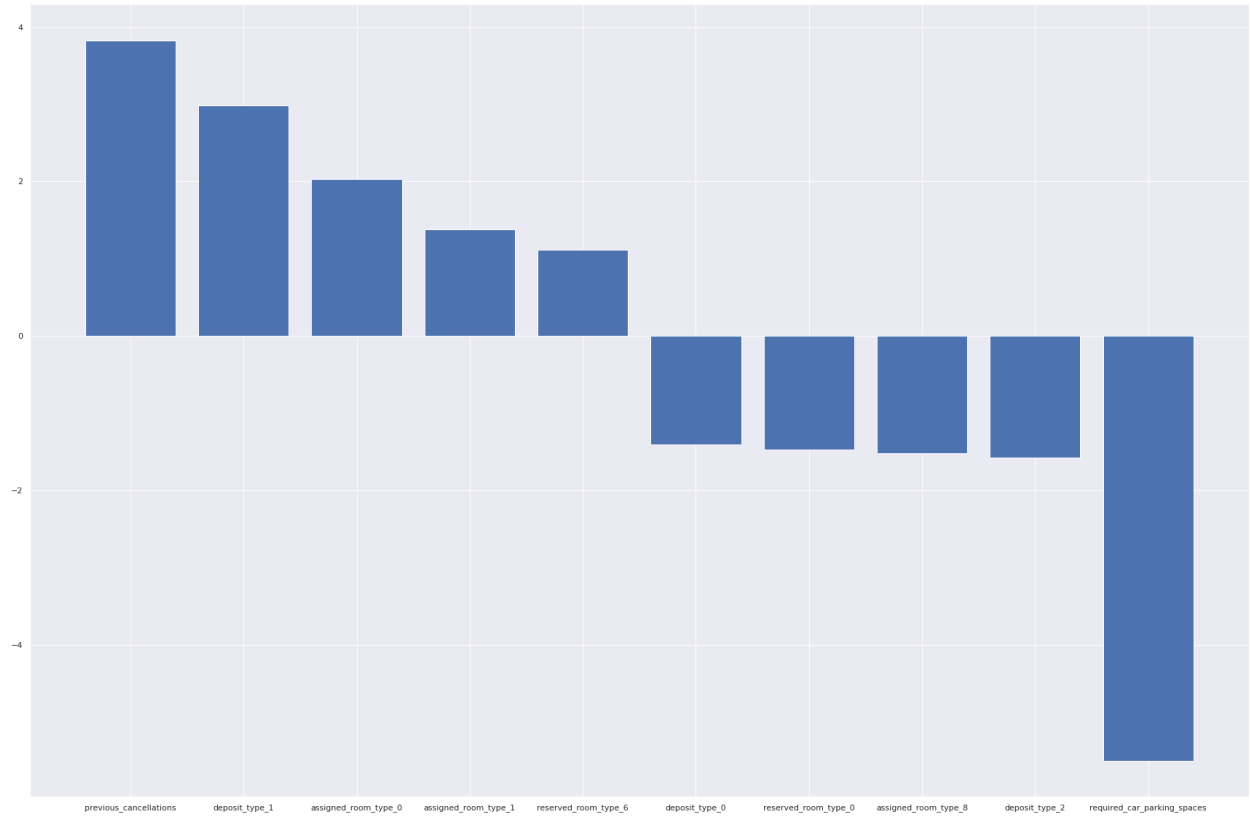
We generate feature importance for each column and test the p-value to select significant contributors.



The ROC curve looks decent, giving an AUC score of 0.90. This score implies that our model is a good fit.

The accuracy this model achieved was 81%. It is indeed not a bad accuracy for a simple model.

Let's look at the feature importance graph for this model.



This graph shows the top 5 positive importances and top 5 negative importances. "previous_cancellations" is the top positive importance. Clearly, it means that if a person has higher values of "previous_cancellations", they have a higher chance of canceling their bookings in the future. Conversely, in the case of the top negative importance "required_car_parking_spaces", a higher value (yes) will imply a lower chance of canceling the booking.

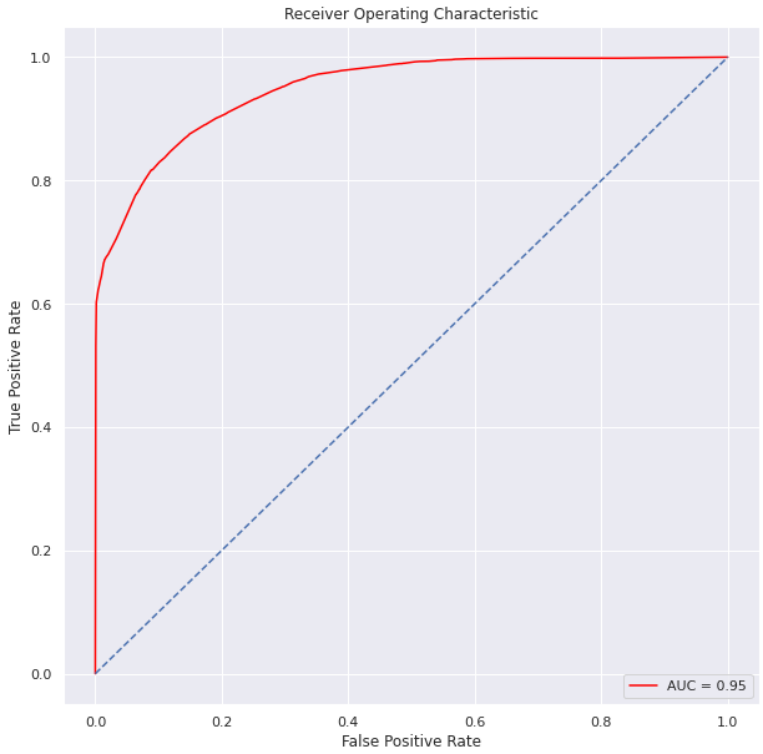
It will be interesting to see the feature importances for our best model, Random forest, later.

Decision Tree

Decision tree classifiers are used successfully in many diverse areas.

Their salient feature is their ability to capture descriptive decision-making knowledge from the supplied data.

Feature importance can be explored from this algorithm, but if we do the same for every model, it will add unnecessary bulk to this report. Therefore, we will only explore the feature importance of our best model, spoiler alert, the Random Forest.



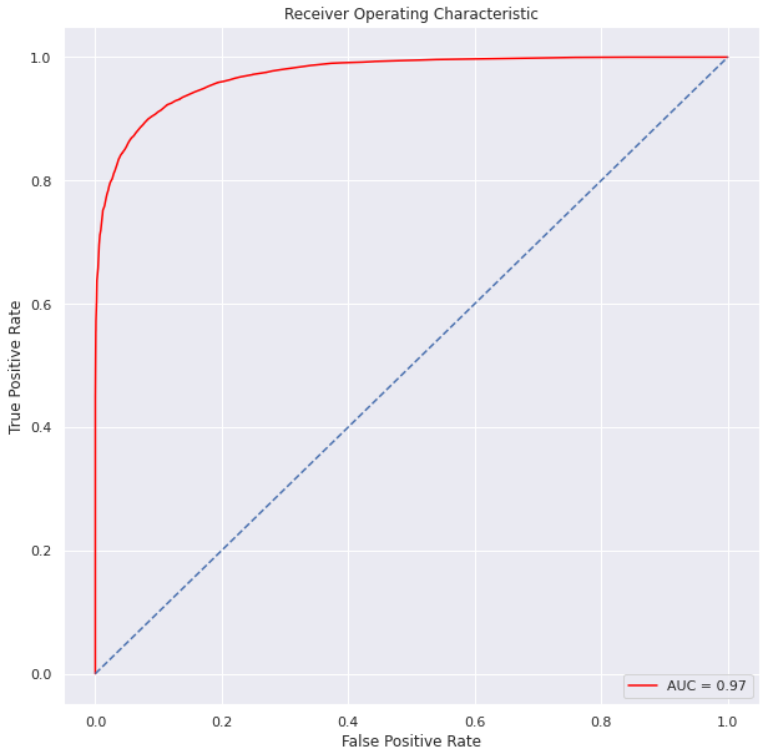
The ROC curve is an even better fit and gives us an AUC score of 0.95.

The accuracy that we achieved was 86.4%. Which is slightly better than what we expected based on the ROC-AUC curve.

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

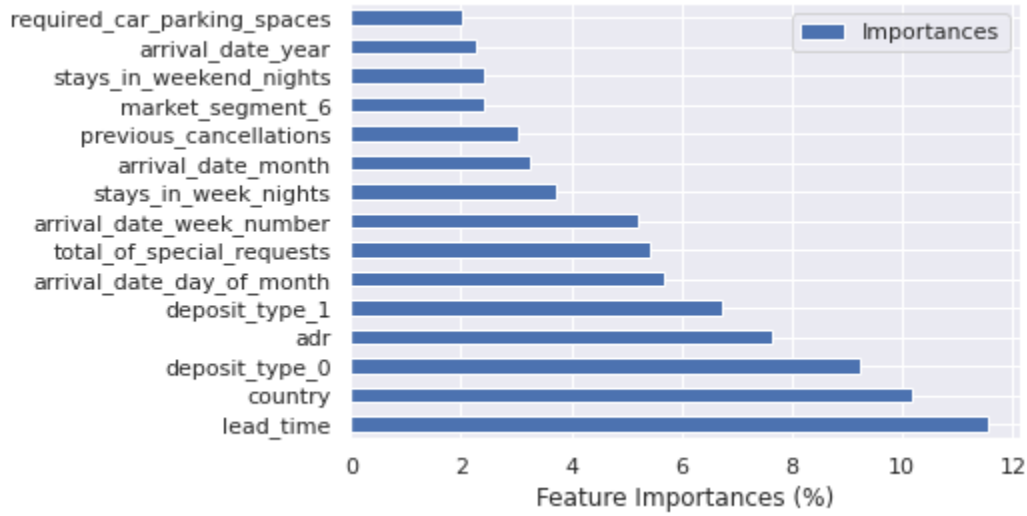
One of the most critical features of the Random Forest Algorithm is that it can handle data sets containing continuous variables as in the case of regression and categorical variables as in the case of classification. Here we will be using it for classification.



The ROC curve is exceptionally good and gives us an AUC score of 0.97.

The accuracy that we achieved was 90.8%. This is amazingly accurate. Interestingly, we note that just an increase of 0.02 area under the curve results in an accuracy increase of about 4%.

Now we will look at the feature importance.



Unfortunately, we were unable to extract the importances in a manner that showed positive/negative correlations for each feature.

We have several intriguing observations from this graphic. First, a quick summary of the three most important features:

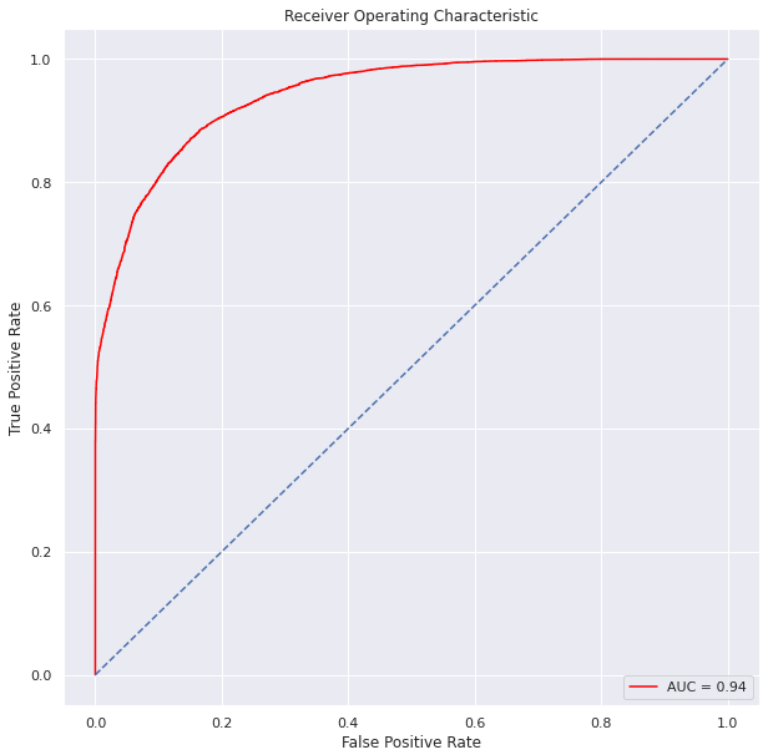
1. lead_time: Number of days that elapsed between the entering date of the booking into the PMS and the arrival date.
2. country: Country of origin of the customer.
3. deposit_type_1: Non Refund – a deposit was made in the value of the total stay cost.

Now, our observations:

- Behold, the "country" column redeems itself splendidly. We are excited to see our efforts bearing fruit.
- The most "important feature" in this model is "lead_time". We will look more closely into this shortly.
- Deposit type is the only feature common in the top 5 for both logistic regression and random forest. Clearly, people do not wish to lose their deposits to Hotels.

XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGB is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving structured data XGBoost and tree-based algorithms are considered superior choices. We expected a decent competition between this algorithm and Random Forest, but it was not to be.



The accuracy achieved here was 85.8%. The Roc curve shows a slightly lesser fit than the decision tree algorithm at AUC 0.94.

Due to the gradient boosting, the feature importance of this method is also not reliable. Thus our suggestion is to avoid XGBoost unless the accuracies achieved are significantly greater than the ones achieved in other models.

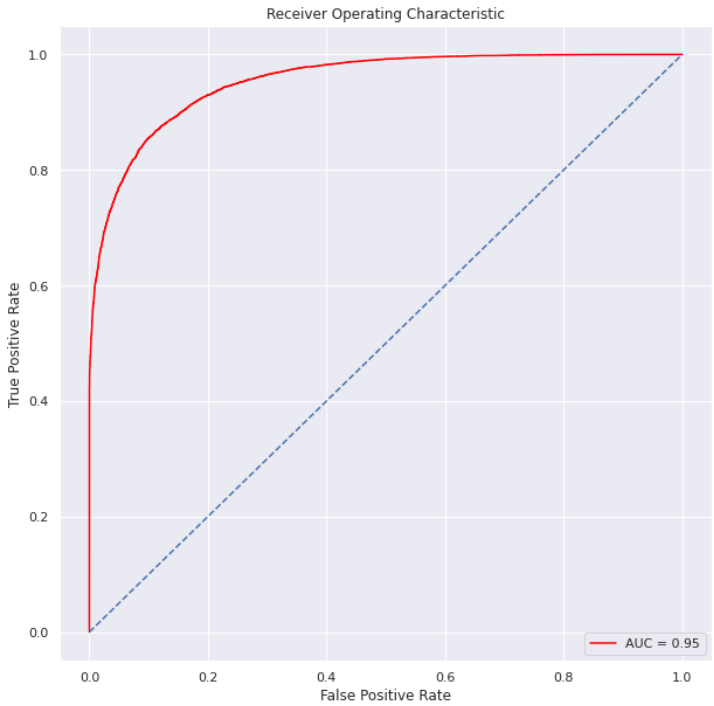
Neural Network MLP

The field of artificial neural networks is often just called neural networks or multi-layer perceptrons, after perhaps the most helpful type of neural network. A perceptron is a single neuron model that was a precursor to more extensive neural networks.

Neural networks learn the representation of training data by adjusting the weights and calculation errors multiple times; they can learn mapping functions with great accuracy and are a universal approximation algorithm.

The predictive capability comes from the multi-layered structure. The weights it takes and the neural connection based on them help it in different scales and combine them into higher-order and more complex functions.

We use the MLP classifier and tune the hyperparameters using grid search for the best accuracies.



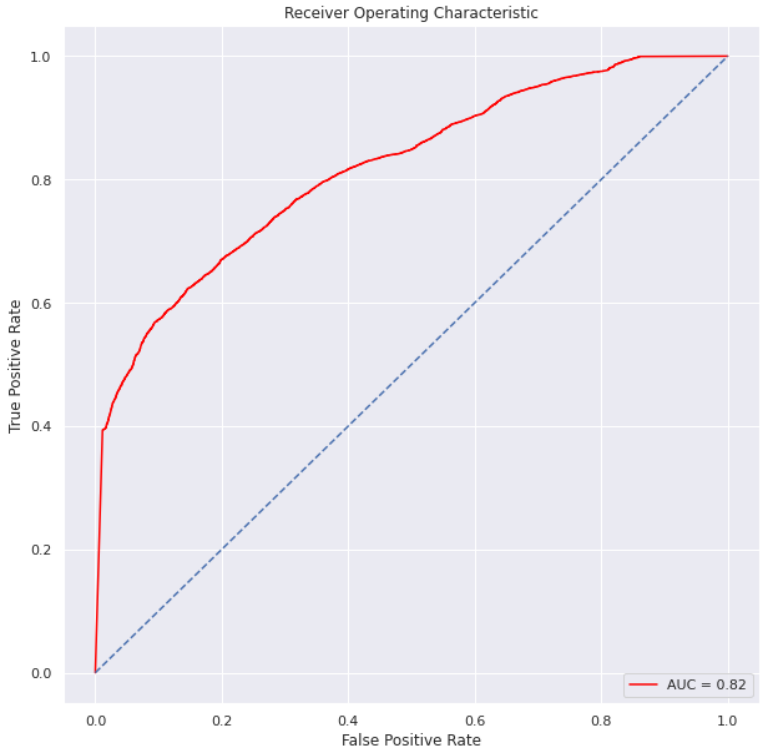
We notice that the ROC curve is similar to the decision tree and the accuracy too is similar, we achieved an accuracy of 85.9% which is good but here we lose the complete interpretability of our feature since this is an unsupervised learning algorithm. From a business perspective, it would be suggested to not use this until it yields exceptionally better results.

Naive Bayes

Naive Bayes classifier is a technique where the classifications are made based on the assumption that the features are independent.

Our data has correlated features, and not every feature is independent; thus, the fundamental rule behind this algorithm is flawed for our prediction, but we go ahead and still test this algorithm as it is the simplest classification algorithm. Further, multiple features add noise to our model.

The results are poor as expected, and the ROC curve does not show a good fit either.



The accuracy achieved was 59% which is extremely poor compared to our other models.

RESULTS SUMMARY

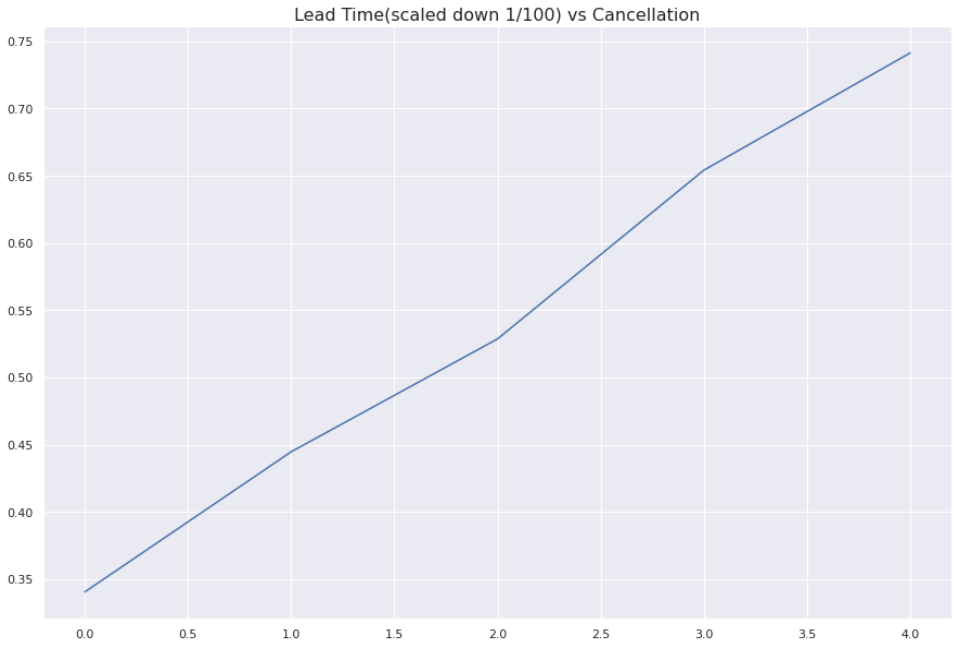
Model	Accuracy (Imbalanced)	Accuracy (Balanced)	ROC AUC Score (Bal)	ROC AUC Score (Imb)
Random Forest	0.883	0.908	0.95	0.97
Neural Network Tuned	0.859	0.885	0.94	0.96
Neural Network	0.849	0.877	0.93	0.95
Decision Tree	0.834	0.864	0.91	0.95
XGBoost	0.838	0.858	0.92	0.94
Logistic Regression	0.802	0.810	0.87	0.90
Naive Bayes	0.590	0.678	0.78	0.82

We decided to train our models on the imbalanced data as well to see how that affects the accuracy since, in the real world, data will always be imbalanced, and there will be fewer cancellations than bookings.

We see that the models trained on balanced data perform better than those trained on imbalanced data. This indicates that the models always train better with balanced data and make better predictions irrespective of the real-world imbalance in data, a fascinating observation.

A sojourn into “lead_time”

As mentioned in the previous section, we decide to dig a little deeper into the most important feature, "lead_time".



We see a result that is initially surprising, but not much so if thought in hindsight. It seems that "is_canceled", our target variable, has a rather linear relationship with the most important feature, "lead_time". Clearly, bookings done long in advance seem to have a correspondingly higher chance of cancellation. Similarly, bookings done on short notice are less likely to be canceled—another fascinating insight.

CONCLUSION

As is clear from the model summary, Random Forest performed best. Besides being the best performer, the feature importance graph indicates that the model takes, arguably, the best features into account while making predictions as we, based on logic, believe them to be the most crucial real-world features of the entire lot.

In fact, the accuracy is good enough that with some improvements, as we state in the next section, we believe this model can be used industrially; to ration for the cancellations in their projections of revenue and sales.

IMPROVEMENTS AND NEXT STEPS

Despite the excellent accuracy of our best-performing model, there is room for some improvements:

- ☐ Some feature selection and feature engineering techniques. It is predicted that by 2030 more than 50% of the data when building models will be synthetic data.
- ☐ Better data collection strategies. Collecting good data is as essential as making prediction models. (e.g., price range bins, we can predict the number of cancellations per bin to further the analysis)

FINAL WORDS

As we have discussed, cancellations are a huge problem for the Hotel Industry. They can impact the sales to a great value.

Accurately predicting whether a customer is likely to cancel their booking at the time of booking itself, the business will be able to accommodate accordingly. (e.g. by overbooking to a certain extent). This will lead to a growth in sales and revenue for these businesses.

DATA SOURCE

https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand?datasetId=511638&sortBy=voteCount