

Machine Learning & Data Mining

Final Project

**Hotel Booking Cancellations**

Professor: Boaz Lerner

Students: Oriel Perets & Dafna Meron

Date: 21.02.22

**Abstract**

In the one and a half trillion dollars hotel industry, demand management is a crucial factor to revenue. Hoteliers’ ability to make accurate demand forecasting is greatly affected by booking cancellations, which often result in vacancy, inaccurate room pricing, false cancellation policy implementation resulting in further reduction of the number of bookings and revenue. Accurate demand forecasting also assures hoteliers and reservation giants (such as Booking and Trivago) correct use of over-booking tactics, incentive-based programs, room deposits and cancellation policies. In this project we use data collected for over 119,000 bookings, combined with interpretable machine learning based classifiers to provide a tool which will be able to accurately predict booking cancellations in advance. The use of interpretable models (specifically decision tree-based models) will provide hotel management with important insights, as well as the ability to base management level decision on understandable cancellation and demand patterns. In this project, we demonstrate a Random Forest Classifier with 88% Recall score.

Contents

[Business Understanding 4](#_Toc96290044)

[Data Understanding 4](#_Toc96290045)

[Collect data 4](#_Toc96290046)

[Describe data 5](#_Toc96290047)

[Explore data 5](#_Toc96290048)

[Data Preparation 6](#_Toc96290049)

[Select data 6](#_Toc96290050)

[Clean data 7](#_Toc96290051)

[Construct data 7](#_Toc96290052)

[Format data 8](#_Toc96290053)

[Modeling 9](#_Toc96290054)

[Models tested 9](#_Toc96290055)

[Validation method 9](#_Toc96290056)

[Feature Importance 9](#_Toc96290057)

[Hyper parameter tuning 10](#_Toc96290058)

[Evaluation 10](#_Toc96290059)

[Models comparison – default parameters 10](#_Toc96290060)

[Models comparison – tuned parameters 11](#_Toc96290061)

[Review process 12](#_Toc96290062)

[Discussion and Conclusions 12](#_Toc96290063)

[References 13](#_Toc96290064)

[Appendices 14](#_Toc96290065)

## Business Understanding

#### Industry Overview

The hospitality industry is a 4.1 trillion dollar a year industry as of 2021(Statista.com) – Comprising of Hotels, Amusement parks, lodging, food and drinks services, travel, tourism and more. The hotel sector alone comprises of 35% of the global industry, averaging 1.5 trillion dollars a year.

#### Business Objective

In the hotel industry, demand forecasting and modeling is a crucial factor to revenue. Hoteliers’ and accommodation websites’ objective is primarily maximizing revenue. This objective can be achieved using accurate demand forecasting, room pricing and cancellation policies accordingly. By doing that, hoteliers aim to maximize occupancy and minimize vacancy of their hotels, while accommodation websites aim to maximize fulfilled bookings (total bookings excluding booking cancellations).

#### Situation Assessment

Accurate demand forecasting is highly impacted by booking cancellations, causing uncertainty around demand management decisions. To mitigate booking cancellations, hoteliers and accommodation websites may employ strict cancellation policies or overbooking tactics which in turn reduces the number of bookings, reduce revenue (N. Antonio, A. Almeida, L. Nunes, 2019) and making it an ineffective solution.

Based on the data collected in this work, over 35% of bookings end up being cancelled, 50% of which (or 17.5% of all bookings) are cancelled within the 70 days (t-70) prior to the arrival date. 25% of cancelled bookings (or 8.75% of all bookings) are cancelled within the 18 days (t-18) prior to the arrival date. This makes it increasingly difficult to accurately predict demand in advance and showcases the increasing need for a dynamic, accurate cancellation forecasting model.

In recent years, as uncertainty rises due to the COVID-19 pandemic and rapid changing restrictions, booking cancellations are increasingly affecting hotels’ and accommodation websites’ demand forecasting techniques by introducing new, often very hard to predict factors to the customer’s booking behavior.

Booking cancellation policies and penalties are important factors in customer booking decisions (C. Chen, Z. Schwartz, P. Vargas). Accurate booking cancellation forecasting can assist hoteliers and accommodation websites accurately price and allocate booking cancellation policies, per customer, based on historic actions and current booking properties.

#### Our Solution

We aim to use the data collected, to train a machine learning model, which can predict a booking cancellation given all or some of the above features, with sufficient accuracy (more specifically, recall score).

## Data Understanding

### Collect data

The data was collected from Kaggle.com, “[Hotel Booking Demand](https://www.kaggle.com/jessemostipak/hotel-booking-demand)”.

### Describe data

The dataset contains 119,390 rows and 33 columns. It was collected as a csv file – each row represents a single booking instance, described by 30 distinct attributes. As part of the data understanding process, we learn that all attributes can be categorized into six categories. Each category describes a certain aspect of at booking instance – Time, Guest, Reservation, History, Channel and Other. Figure 1 presents some attribute examples in each category.

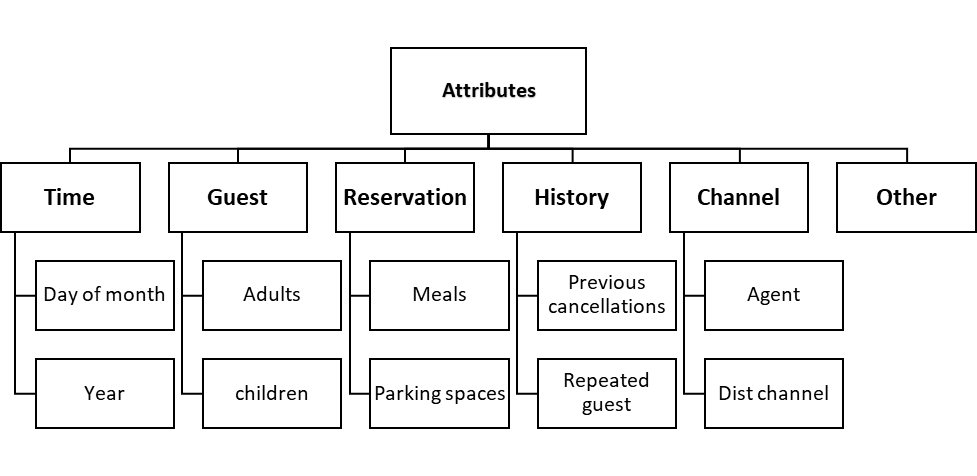


Figure 1

Following the categorization above, we performed specific attribute description and understanding, including descriptive statistics calculations for categorical variables (table 8) and numeric variables (table 9). This description can be found in the appendices at the end of this report.

### Explore data

We consider the target variable ‘is\_canceled’ to be balanced, distributing 37% - canceled bookings and 63% fulfilled bookings.

As part of the data understanding process, we chose to illustrate interesting relationships and correlations between several attributes in the dataset and the target variable or its derivatives. In figures 2, we present the percentage of canceled bookings in several attributes. In figure 3, we present relationship between the number of previous cancellations and special requests by customer.

Some interesting conclusions include:

* Full Board is the meal plan with the highest cancellation percentage, 60%. (Figure 2)
* New guests are 20% more likely to cancel a booking when compared to returning guests (Figure 2).
* Guests booking through Travel Agents and Operators are twice as likely to cancel a booking as any other distribution channel (Figure 2).
* Most guests have 0 special requests, however, the more special requests a guest have, the more likely he/she is to cancel the booking (Figure 3).

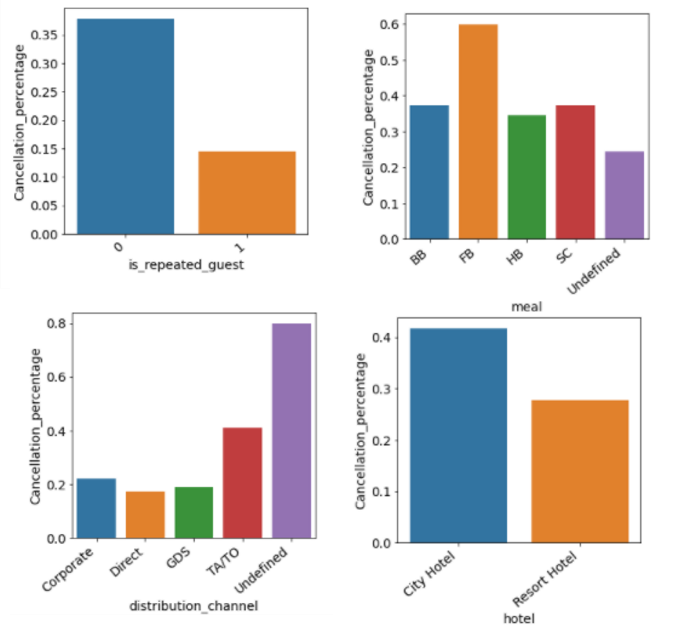


Figure 2



Figure 3

## Data Preparation

### Select data

Our task is to classify canceled and fulfilled bookings. While exploring the data we found two attributes other than ‘is\_canceled’ when combined give similar information:

'reservation\_status\_date’, and 'reservation\_status'

We selected ‘is canceled’ as target variable and removed the other two attributes from our data.

### Clean data

#### Missing values

The number of missing values that were observed in four attributes and the action taken to handle them are presented in table 1:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Number of missing values** | **Action** |
| ‘children’ | 4 | Fill with median |
| ‘country’ | 448 | Fill with ‘unknown’ |
| ‘agent’ | 16,340 | Remove attribute |
| ‘company’ | 112,593 | Remove attribute |

Table 1

Since in ‘children’ and ‘country’ the number of missing values is small compared to the number of entities (119,390), we decided to fill them with the attributes median value. In ‘agent’ and ‘company’ the number was too high, so we decided to remove the entire attribute.

#### Undefined categories

Three categorical attributes: ‘meal’, ‘market\_segment’ and ‘distribution\_channel’ included an ‘undefined’ category. The number of records under this category was less than 1.5% for each attribute. ‘undefined’ was replaced with maximum size category as can be seen in figure

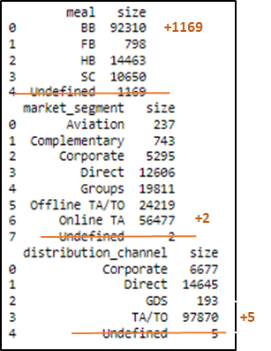


Figure 4

### Construct data

Derived attributes we decided to create, and their computation are presented in table 2:

|  |  |  |
| --- | --- | --- |
| **Derived Attributes** | **Type** | **Computation** |
| ‘Is\_reserved\_equals\_assigned’ | Boolean | ‘reserved\_room\_type’ == ‘assigned\_room\_type’ |
| ‘day\_of\_Year’ | Numeric | timetuple().tm\_yday |
| ‘is\_holiday\_season’ | Boolean | 'day\_of\_year‘ >= 355 |
| ‘total\_stay\_nights’ | Numeric | 'stays\_in\_weekend\_nights‘ + 'stays\_in\_week\_nights' |
| ‘total\_guests’ | Numeric | 'adults‘ + 'children' + 'babies' |

Table 2

‘Is\_reserved\_equals\_assigned’ - describes whether there was a match between the guest requested room and the one that was eventually reserved for them. We believe, cases where the guest did not get the room they requested could be associated with cancellation.

‘day\_of\_Year’ – describes the number of arrival day in the year, from 1 to 365 (or 366 in a leap year like 2016). This attribute was created to widen the time dimension in our data, and help create the next derived attribute.

‘is\_holiday\_season’ - describes whether the reservation arrival day is in the last 10 days of the year - around Charismas and New Year’s Eve. We believe this is a special time in the year, and therefore our target variable may also behave differently.

‘total\_stay\_nights’ - this attribute was created in order to add more data related to the length of stay.

‘total\_guests’- This attribute was created in order add more data related to the party size.

### Format data

#### Categorization by binning

We decided to perform categorization using binning for several numeric attributes. We used the attributes’ distribution to decide how to perform the binning (example in figure 5).

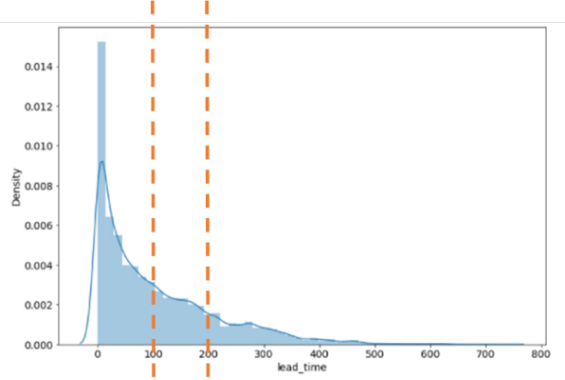


Figure 5

* ‘lead\_time’ was divided to the following intervals: 0-100 | 100-200 | 200+
* ‘adr’ was divided to 4 equal size bins.
* ‘booking\_changes’ was divided to 5 bins according to these ranges: [0,1), [1,2), [2,3), [3,6), [6,∞)
* ‘previous\_cancellations’ was divided to 4 bins according to these ranges: [0,1), [1,2), [2,4), [5,∞)
* ‘required\_car\_parking\_space’ was converted to binary according to these ranges: (-∞,0), [0,∞)

#### Categorization by LabelEncoder()

For the following attributes, we decided to perform categorization using LabelEncoder() which encodes labels such that they contain only values between 0 and n\_classes-1:

‘hotel’, ‘meal’, ‘country’, ‘market\_segment’ , ‘distribution\_channel’, ‘reserved\_room\_type’, ‘assigned\_room\_type’, ‘deposit\_type’, ‘customer\_type’.

## Modeling

### Models tested

Decision Tree based models are the most suitable for our kind of data - tabular and explainable. Therefore, we decided to test Random Forest and XG-Boost. We chose to compare results with two other common models: Logistic Regression and Gaussian-NB.

### Validation method

We used 10-fold cross validation. For each cross-validation loop, the model which gave the best accuracy score was used for further evaluation and comparison. First, we ran each model with its default parameters (table 3).

|  |  |
| --- | --- |
| **Model** | **Default Parameters** |
| RandomForestClassifier | (n\_estimators = 100, max\_features=‘auto’, max\_depth=None) |
| GradientBoostingClassifier | )n\_estimators=100, learning\_rate=0.1, max\_depth=3) |
| LogisticRegression | (penalty='l2‘, max\_iter=100) |
| GaussianNB | ( var\_smoothing=1e-09) |

Table 3

### Feature Importance

For both Random Forest and Gradient Boosting Classifiers, we decided to perform a feature importance analysis and see what we can be inferred from the model’s explain-ability. In this analysis (figure 6), the higher the feature, the more important it is. The importance of a feature is computed as the (normalized) total reduction of the criterion ***{“gini”}*** brought by that feature, also known as the Gini importance.

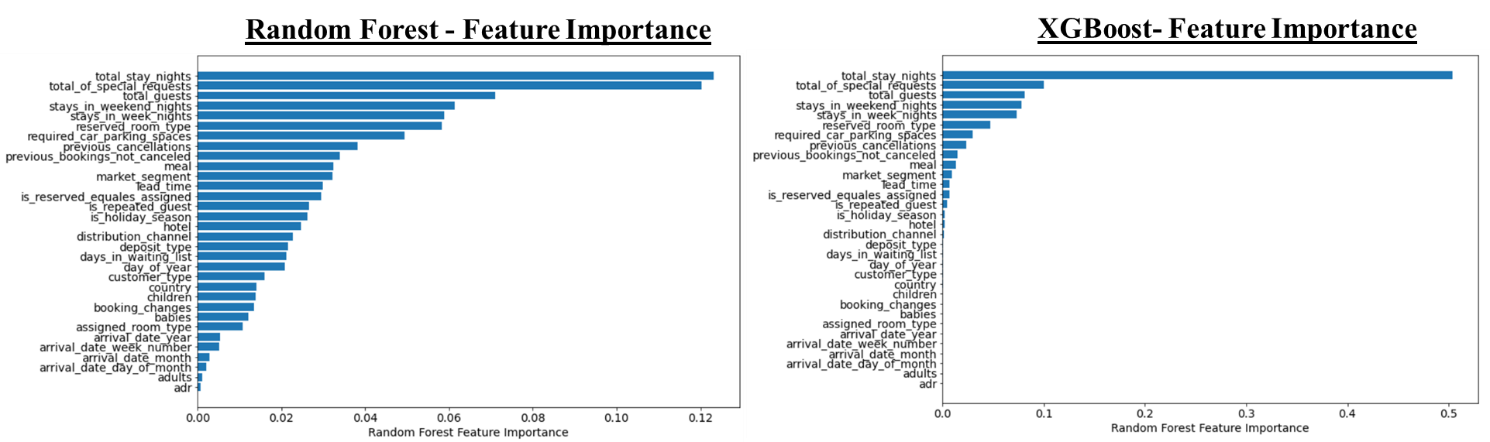


Figure 6

We can observe the feature importance ranking is identical in both models as expected. In XG-Boost we notice more distinguishable differences between strong and weak features. This result is also expected, since in Random Forest each tree is built independently, while in XG-Boost every tree is built considering the errors from the previous tree, considering only the most important features in later stages.

We can notice newly created features ‘total\_stay\_nights’, ‘total\_guests’, ‘is\_reserved\_equals\_assigned’ and ‘is\_holiday\_season’ were ranked with high model importance.

### Hyper parameter tuning

We used grid search cross-validation to perform hyper parameter tuning to all models, except GaussianNB which had no relevant parameters to tune. Presented in table 4.

|  |  |
| --- | --- |
| **Model** | **Tuned Parameters** |
| RandomForestClassifier | param\_grid = {'n\_estimators': [100, 200, 300], 'max\_depth' : [2, 4, None], 'min\_samples\_split': [2, 4]} |
| GradientBoostingClassifier | param\_grid = {'n\_estimators': [50, 100, 200], 'max\_depth' : [2, 4, 8], 'learning\_rate' : [0.05, 0.1]} |
| LogisticRegression | param\_grid = {'max\_iter': [200, 500, 1000], 'penalty': ['l1', 'l2']} |

Table 4

## Evaluation

### Models comparison – default parameters

Table 5 and figure 8 present models’ comparison for the **default parameters** using different scoring metrics. We can observe the Random Forest Classifier provided the best score across all metrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** |
| RandomForestClassifier | 0.89 | 0.88 | **0.87** | 0.88 |
| GradientBoostingClassifier | 0.84 | 0.84 | 0.81 | 0.82 |
| LogisticRegression | 0.76 | 0.75 | 0.73 | 0.74 |
| GaussianNB | 0.60 | 0.70 | 0.67 | 0.60 |

Table 5

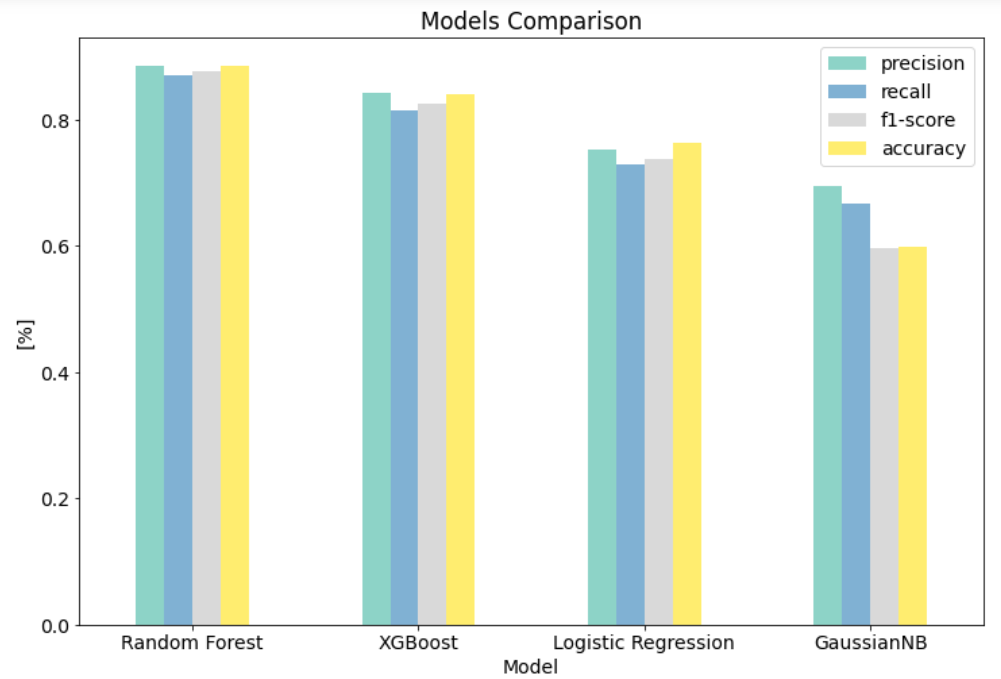


Figure 7

We chose ‘recall’ to evaluate our models. In our case, recall measures the number of cancellations the model successfully identified out of the overall cancellations, but does not measure the number of false cancellations. Recall was used over other metrics based on the business understanding process - hotel bookings are less sensitive to false-positives (a.i predicting a booking cancellation when the booking is fulfilled), as cancellation preparation tactics often require low-cost implementation. Therefore, based on specific management decisions, hotel chains and reservation websites might chose to lower, or raise the classification threshold, based on their specific needs, cancellation tolerance, and desired deposit and cancellation policies.

### Models comparison – tuned parameters

Table 6 presents best parameters for each model. We can identify overfitting by comparing recall test and train scores. Also, we can compare each model’s average training time with its best parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Train** | **Test** | **Best Parameters** | **Avg. Train Time** | **Improved** |
| RandomForestClassifier | **0.958** | **0.88** | {'max\_depth': None, 'max\_features': 'auto', 'n\_estimators': 300} | 1 min | X |
| GradientBoostingClassifier | 0.90 | 0.87 | {'learning\_rate': 0.1, 0.05,'max\_depth': 8, 'n\_estimators': 200} | 1.7 min | V |
| LogisticRegression | 0.79 | 0.79 | {'max\_iter': 1000, 'penalty': 'l2‘} | 25 sec | V |

Table 6

Performing hyper parameter tuning on RandomForestClassifier didn’t improve the recall score, increasing the number of estimators only caused overfitting. For GradientBoostingClassifier, the process resulted in a 4% improvement in recall score, by increasing maximum tree depth to 8 and number of estimators to 200, without reaching over fitting. For LogisticRegression hyper parameter tuning process resulted in 1% improvement in recall score, by increasing maximum iterations to 1000. In general, we observed that most of the improvement came from parameters that control the “size” (or complexity) of the model. We decided to end the tuning at this point, because further tuning would have negligible effect on the score and would take much longer training time while risking in over fitting.

## Review process

#### Binning

In our initial data preparation, we performed categorization by binning, using mainly qualitative methods. After reviewing our process, we decided to use quantitative method of KBinsDiscretizer from sklearn.preprocessing package, which implements different binning strategies. We used the ‘kmeans’ strategy that defines bins based on a k-means clustering procedure performed on each feature independently.

#### Inference from feature importance

Feature importance analysis highlighted some information details in the reservation that are more important to the model. In order to exploit this kind of information we decided to created additional features (table 7):

|  |  |  |
| --- | --- | --- |
| Derived Attributes | Type | Computation |
| ‘cancellation\_precentage’ | Numeric | ‘previous\_cancellations’/ (‘previous\_cancellations’ + (‘previous\_bookings\_not\_canceled’) |
| ‘nights\_times\_lead’ | Numeric | ‘total\_stay\_nights’ \* ‘lead\_time’ |
| ‘nights\_lead\_ratio’ | Numeric | ‘total\_stay\_nights’ / ‘lead\_time’ |

Table 7

‘cancellation\_precentage’ - describes the number of previous cancellations of a costumer divided by their total reservations amount. Looking at this new feature distribution brought us to perform binning to 3 categories, using these ranges: [0,1), [1,2), [2,∞).

‘nights\_times\_lead’ - describes the number of total reserved nights multiplied by the number of days between the date of booking and the date of arrival.

‘nights\_lead\_ratio’ - describes the number of total reserved nights divided by the number of days between the date of booking and the date of arrival.

Changing the binning method and creating these new features, eventually improved improve the classification recall score in both XGBoost and RF by 1%, to 86% and 88% respectively.

## Discussion and Conclusions

The Random Forest Classifier was the overall best performing model, given our specific dataset, training and testing processes. Whereas a Gradient Boosting Classifier might be a better alternative when testing the model with fewer parameters. In the hyper parameter optimization process RandomForest Generated the best score using its default parameters, XGBoost’s score was improved by more estimators and deeper trees, and the logistic regression model improved as we increased iterations. Furthermore, using feature importance analysis to refine the data preparation process proved to be an effective way to increase the model’s score.

## References

1. Chen, C.C., Schwartz, Z. and Vargas, P., 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), pp.129-135.
2. Antonio, N., de Almeida, A. and Nunes, L., 2019. Big data in hotel revenue management: Exploring cancellation drivers to gain insights into booking cancellation behavior. Cornell Hospitality Quarterly, 60(4), pp.298-319.
3. Antonio, N., De Almeida, A. and Nunes, L., 2017. Predicting hotel booking cancellations to decrease uncertainty and increase revenue. Tourism & Management Studies, 13(2), pp.25-39.

## Appendices

#### Categorical variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature Name | Type | Top | Values | Explanation |
| Hotel | String | City Hotel | [Resort, City] | Represents the type of the hotel, Resort or City |
| Is\_canceled | Boolean | 0 | [0, 1] | Was the booking canceled |
| Arrival\_date\_year | Int | 2016 | [2015,2016,2017] | The year of the hotel arrival date |
| Arrival\_date\_month | String | August | [January, …, December] | The month of the hotel arrival date |
| Arrival\_date\_week\_number | Int | 33 | (1,52) | The week of the year of the hotel arrival date |
| Arrival\_date\_day\_of\_month | Int | 17 | (0,31) | The day of the month of the hotel arrival date |
| Meal | String | BB | [BB, HB, FB, SC, Undefined] | The accommodation meal plan |
| Country | String | PRT | 177 Countries | The Country of the hotel |
| Market\_segment | String |  | [Online TA, Offline TA/TO, Groups, Direct, Corporate, Complementary, Aviation, Undefined] | Market Segment – TA (travel agent) or TO (tour operator) |
| Distribution\_channel | String | TA/TO | [TA/TO, Direct, Corporate, DGS, Undefined] | Distribution Channel, TA (travel agent), TO(tour operator) |
| Is\_repeated\_guest | Boolean | 0 - New | [0,1] | Is the guest a repeat customer |
| Reserved\_room\_type | String | A | [A,B,C,D,E,F,G,H,P,L] | The type of room reserved by the customer |
| Assignmed\_room\_type | String | A | [A,B,C,D,E,F,G,H,P,L] | The type of room assigned to the customer |
| Deposit\_type | String | No deposit | [No Deposit, Refundable, Non-refundable | The type of deposit made by the customer |
| Company | Int | 40 | (6,543) | ID representing the company which paid for the booking or booked the rooms for the guests |
| Agent | Float | 9 | (1,535) | ID representing the travel agency the booking came from |
| Customer\_type | String | Transient | [Transient, Transient-party, Contract, Group] | Type of the customer who booked |

Table 8

#### Numeric Variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature Name | Mean | SD | Values | Explanation |
| Lead\_time | 104.01 | 106.86 | (0, 463) | Days before hotel arrival date |
| Stays\_in\_weekend\_nights | 2.50 | 1.90 | (0,19) | # of weekend nights in the booking |
| Stays\_in\_week\_nights | 0.92 | 0.99 | (0,50) | # of weekday nights in the booking |
| Adults | 1.85 | 0.58 | (0,55) | # of adults in the booking |
| Children | 0.10 | 0.39 | (0,10) | # of children in the booking |
| Babies | 0.007 | 0.09 | (0,10) | # of babies in the booking |
| Previous\_cancellation | 0.08 | 0.84 | (0,26) | # of previous cancellations for the guest |
| Previous\_not\_cancaled | 0.13 | 1.49 | (0,72) | # of previous bookings not canceled |
| Booking\_changes | 0.22 | 0.65 | (0,21) | # of booking changes done by the customer |
| Days\_in\_waiting\_list | 2.32 | 17.59 | (0,391) | Days the customer has waiting in the waiting list for a room |
| Adr | 101.83 | 50.53 | (62, 157) | The average daily rate |
| Required\_car\_parking\_spacs | 0.06 | 0.24 | (0,8) | # of parking spaces the customer asked for |
| Total\_special\_requests | 0.57 | 0.79 | (0,5) | # of special requests asked |

Table 9