BC2: Predict Hotel Booking Cancellations

MASTER'S DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

March 2022

Q Consulting

Laura Isabella Cuna, 20211312

Amelie Florentine Langenstein, 20210637

Tongjiuzhou Liu, 20211012

Nina Urbancic, m20211314

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Table of Contents

1. Introduction

With the appearance of Online Travel Agencies, the hotel industry is facing major issues with cancellations of bookings. Loose cancellation policies allow "deal-seeking" customers to make multiple bookings or make one booking and continue to search for better deals, and cancel when they find it. The competition intensified, as it is much easier to compare prices and services of different hotels presented by online travel agencies. This results in high cancellation rates and major reductions in revenue.

Hotel chain C has been struggling with the same issue, and has hired us to reduce the uncertainty regarding the actual demand of rooms to avoid overbooking as well as leaving rooms empty. They assigned us to develop a model to predict the net demand for their hotel H2, based on a dataset of hotel bookings between July 1, 2015 and August 31st, 2017.

According to the CRISP-DM methodology, we first elaborated the business objectives and set the data mining goal for this project. Then we moved on to exploring and understanding the dataset and performing data cleaning and transformation necessary for the modeling part. We applied and compared different machine learning algorithms and selected the best performing model to our unseen data.

2. Business Understanding

Hotel chain C has seen a severe negative impact caused by cancellations. Between July 2015 and August 2017, around 42% of all bookings were canceled, which let the hotel miss out on almost 11 million euros in revenue. Also, high cancellations result in opportunity costs for vacant rooms. Overbooking is a measure to address this and increase revenue when taking net demand into consideration. However, wrong estimations of net demand lead to unsatisfied customers who have to be moved to a different hotel. This will generate relocation costs, decrease immediate and future revenue and can result in bad reviews from these customers. Tightening cancellation policies would make offers less attractive, decrease the demand and should therefore also not be an option.

In order to implement better pricing and overbooking policies, it is imperative to identify which future customers are more likely to cancel their reservations. This will allow for a more precise estimate of net demand and adjusting special offers and overbooking measures accordingly. This will minimize costs and maximize revenue, while maintaining the reputation of the hotel.

Business Objective

The objective of Hotel Chain C is to reduce cancellations by 20%, which will help to estimate net demand more precisely and thereby avoid vacant rooms, but also reduce the risks of overbooking.

Data Mining Goal

The aim of this data mining project is to predict which future bookings are likely to be canceled and which are not, based on the booking information of hotel H2 from July 2015 to August 2017.

3. Data Understanding

The dataset we received consisted of 79.330 records represented by 31 features. We set the binary feature *IsCanceled* as the target feature. It showed whether the booking was canceled or not and is to be predicted by a model based on a selection of the other 30 features in the dataset. The dataset was slightly unbalanced as 41.7% of the records contained canceled bookings and 58.3% contained not-canceled bookings.

The features in our dataset could be grouped into three different subsets: **date features**, **categorical features** (a category or type) and **numerical features** (where the measurement or number has a numerical meaning). Before modeling, we converted our categorical features into integer format by encoding them with OneHotEncoder which makes our training data more useful and expressive.

We observed 4 missing values in *Children*, and 24 missing values in *Country*. The dataset also contained 31.748 duplicate records. This could be due to multiple bookings of deal-seekers using different OTAs.

We divided the data into **metric** and **non-metric** features for further exploration. We checked for outliers in metric features with the help of box plots and histograms, and in non-metric features using absolute and relative frequencies. We noticed outliers in both types of features.

We also checked for any correlation between the features we were given using the Spearman Correlation Matrix, since it is used to analyze quantitative data which do not obey normality. Based on the histograms we plotted, we saw that only *ArrivalDateWeekNumber* obeyed normality.

We built a Customer Distribution Map to see which countries our customers were coming from.

During our exploratory analysis we also found some nonsense data that was not realistically possible. Most notable were reservations made for babies and children unaccompanied by adults, reservations for zero guests, and reservations where the Average Daily Rate was unrealistically high.

4. Data Preparation

We started with data preparation by dropping all the duplicates (31.748 rows). This left us with 47.582 unique rows.

We dropped *DistributionChannel*, *ReservationStatus*, and *ReservationStatusDate* since they were not relevant for our predictive model. The information from *DistributionChannel* was already contained in *MarketSegment*, and the *ReservationStatus* and *ReservationStatusDate* were only updated after the booking so they did not show the future data that we were predicting.

Considering we found some - but few - missing values in two of the features, we decided to treat them. The missing values in *Children* were substituted with value 0 which was the mode value, and the missing values in *Country* were also substituted with the mode value which was Portugal.

We trimmed the outliers by defining some logical guidelines/thresholds. We dropped all rows where a room was reserved with 8 or more babies, all rows where ADR was 1000€ and above, and all rows where the room was reserved for babies without adults present. We dropped reservations where children were alone and not a part of another booking (transient party), reservations where the total number of guests was 0, and all rows where a customer had been a repeated guest but did not have any previous cancellations nor had not canceled any bookings at the same time.

We created the following new features to support further analysis:

- ***TotalStayNights*** was created by summing *StaysInWeekendNights* and *StaysInWeekNights,*
- ***TotalGuestsNumber*** was created by summing *Adults*, *Children* and *Babies*,
- ***ReservedRoomChanged*** was created by checking whether *ReservedRoomType* and *AssignedRoomType* were the same
- ***TotalPreviousCancellationsRate*** was created by checking the ratio between *PreviousCancellations* and total stays (*PreviousCancellations*+*PreviousBookingsNotCanceled*).

The format of *ArrivalDateMonth* was changed to month number instead of month name.

We created binary features for: customers who only stay on the weekends (*WeekendOnly*), customers with dependents (*Dependents*), those who require parking (*RequiresParking*), and for Portuguese and Non-Portuguese customers (*Portuguese*).

We grouped the customers in 3 different groups based on who they were traveling with: Single, Couple, or Family.

We dropped *ArrivalDateYear*, *StaysInWeekendNights*, *StaysInWeekNights*, *Adults*, *Children*, *Babies*, *Country*, *ReservedRoomType*, *AssignedRoomType*, *RequiredCarParkingSpaces* because their information was represented by the newly created features.

Since our data had many outliers, using the mean and variance of the data for normalization would not work. So we normalized the data using RobustScaler that decentered the data based on the median or quartiles.

We plotted the distribution for numerical and categorical features again with boxplots and histograms to see how our data was distributed after the data preparation.

To see whether our features had any correlation, we used the Spearman correlation coefficient again. It showed us that *ArrivalDateMonth* and *ArrivalDateWeekNumber* were the only two features which were (perfectly) correlated. We dropped *ArrivalDataMonth*.

We also checked the correlation rate between our metric features and our target. The top three features with the highest correlation were *LeadTime* (with the rate of 0.215), *TotalOfSpecialRequests* (with the rate of -0.177), and *TotalPreviousCancellationsRate* (with the rate of -0.128).

Finally, we encoded the categorical features using the OneHotEncoder.

5. Modeling

Considering our data was ordered with time series, we started by reordering the data randomly to avoid the influence of time. After that, we divided the data into training and testing datasets.

Model Selection

We used cross-validation to select the best model. We also tried TimeSeriesSplit, using the past train set to build the model and then predicting on the future test set, but since our data was not on a completely continuous timeline our result was not ideal. Therefore, we randomly rearranged all the data, removing *Year* to break the effect of time on our predictions, and since we found that the cancellations for each month and week are different, they could be used as our features.

We tested the following six models: GaussianNB, LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier. Even though RandomForestClassifier was not the fastest model we tried (running time of the model was 4m 18s), it had the highest F1 score and the highest balanced accuracy. Based this, we decided to use it for our modeling.

We also considered using Self Organizing Maps and Neural Networks but they took too long to run so we discarded them as they were not optimal solutions.

Feature Selection

We selected our features using the function inside the RandomForestClassifier called feature_importance and selected the features where the importance score was higher than 1% (18 features from 542 features total).

Tuning the Hyper-parameters of the Estimator

We generate the out-of-bag score (the accuracy of the trees in the RandomForestClassifier) and the AUC score (area under the ROC curve) before we tuned the hyper-parameters. The output showed that these two scores were already very high. Based on this, we were able to say that the default parameter fit of RandomForestClassifier was better for this example.

We worked on optimizing the parameters by tuning the hyper-parameters of the estimator with GridSearchCV.

In the order of their importance, the parameter test helped us identify the best number of estimators which was 130 in our case, the maximum depth of our decision tree was 13, with the minimum number of samples split 80, the minimum number of leaves 10, and the maximum number of features 9.

6. Evaluation

We plotted the ROC curve with the parameters we defined as best in the modeling part.

We also plotted the Confusion matrix, which helped us understand the performance of our classification. With the help of the matrix we saw that our classification had 6687 true positives (TP) values, 653 false positives (FP), 1305 false negatives (FN), and 1945 true negatives (TN).

Evaluation Criteria

We considered our business objective when selecting the algorithm. The two most important criteria for our choice of the model were the running time efficiency and the score of the results. We identified Precision as the most important score.

We reasoned this by the following: The predictive model will result in some falsely predicted values. However, we should rather avoid False Positives (not canceled, labeled as canceled) than False Negatives (canceled, labeled as not canceled). Therefore, the goal was to increase the Precision value as much as possible, even at the expense of decreasing the Recall. Extremes were not desirable, and we aimed for both Precision and Recall to be as high as possible, i.e. F1 should be as high as possible.

7. Deployment

Our model can be used in the future when the hotel receives a new order to predict the cancellation rate of the order. This way the hotel managers can better estimate net demand and adjust the overbooking range accordingly to ensure the hotel's maximum revenue. Also, if there are bookings that show a high probability to be canceled, the hotel can take measures to prevent the customers from canceling by offering the customer suitable benefits.

When building the model we found that the most important thing was not the complexity of the model but the efficiency. A very simple model could be more accurate than a very complex model. The model will be applied on frequently updated data and should not take too much computing time. Furthermore, daily business decisions that can have a major influence on the hotels revenue streams will be based on the model. Therefore, the model should have low maintenance needs and a low error rate.

To make the model applicable for the regular use within the hotel business context, we suggest a model that updates the data on a daily basis. Furthermore, the model should be fed with new training data on a monthly basis to gain more knowledge on cancellation habits and increase the precision of our model. This improves estimates of net demand and avoids vacant rooms or overbooking. Also, it can support the decision process concerning necessary actions. Regular info on cancellation chances help the hotel to resell vacant rooms on time. Also, the hotel can react before customers actually decide on canceling their booking and make special offers.