Reserved or Cancelled?

Predicting whether a customer will cancel the reservation made at a hotel.

An interesting thing about the ml models is its ability to work in any field and be used for games or stocks or simple questions like our reservation cancellation problem. we will be going through a project of predicting whether a customer will cancel a reservation they have made at a hotel or not. This will help hotels and such businesses in the hospitality industry to decrease uncertainty be well prepared and better their revenue management based on sales.

Data

We get our data from an open-source website Kaggle, and we notice that there are several columns in the data. we call these columns features and the final column our target. target is what we aim to find, in our case, booking status. we see that there are features like no of children, no if room, no of weeknight, booking status etc.

Exploratory Data Analysis

We begin our EDA by looking at the information of the dataset by using a subset of the whole data called the test set. we can use the *.info() function* and it will give us an output like so:

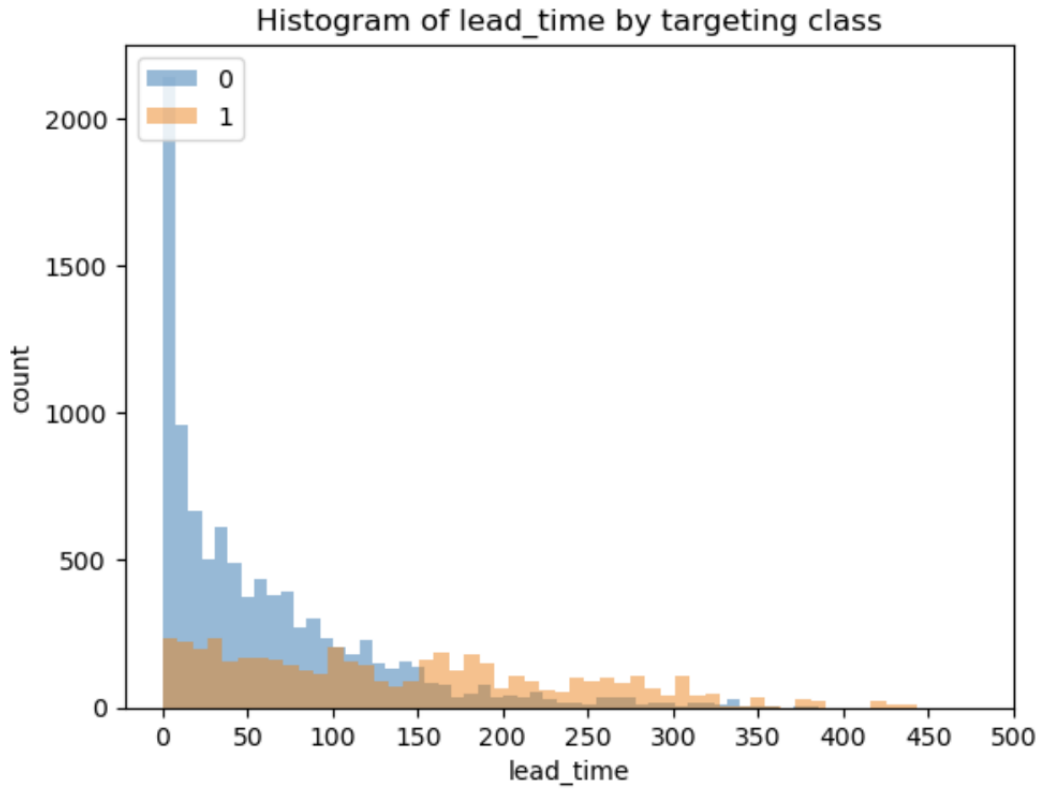
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14509 entries, 14359 to 6682
Data columns (total 20 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   no_of_adults                          14509 non-null  int64
 1   no_of_children                        14509 non-null  int64
 2   no_of_weekend_nights                  14509 non-null  int64
 3   no_of_week_nights                     14509 non-null  int64
 4   type_of_meal_plan                     14509 non-null  int64
 5   required_car_parking_space            14509 non-null  int64
 6   room_type_reserved                    14509 non-null  int64
 7   lead_time                             14509 non-null  int64
 8   arrival_year                          14509 non-null  int64
 9   arrival_month                         14509 non-null  int64
 10  arrival_date                          14509 non-null  int64
 11  market_segment_type                   14509 non-null  int64
 12  repeated_guest                        14509 non-null  int64
 13  no_of_previous_cancellations          14509 non-null  int64
 14  no_of_previous_bookings_not_canceled  14509 non-null  int64
 15  avg_price_per_room                    14509 non-null  float64
 16  no_of_special_requests                14509 non-null  int64
 17  booking_status                        14509 non-null  int64
 18  total_bookings                        14509 non-null  int64
 19  is_holiday_season                     14509 non-null  int64
dtypes: float64(1), int64(19)
memory usage: 2.3 MB
```

from the above image we see that the data in all the features is not null and of the type integer with an exception of *avg_price_per_room* which is of type float.

Fortunately, for us, this dataset did not have any missing pieces, so **no imputation is required**.

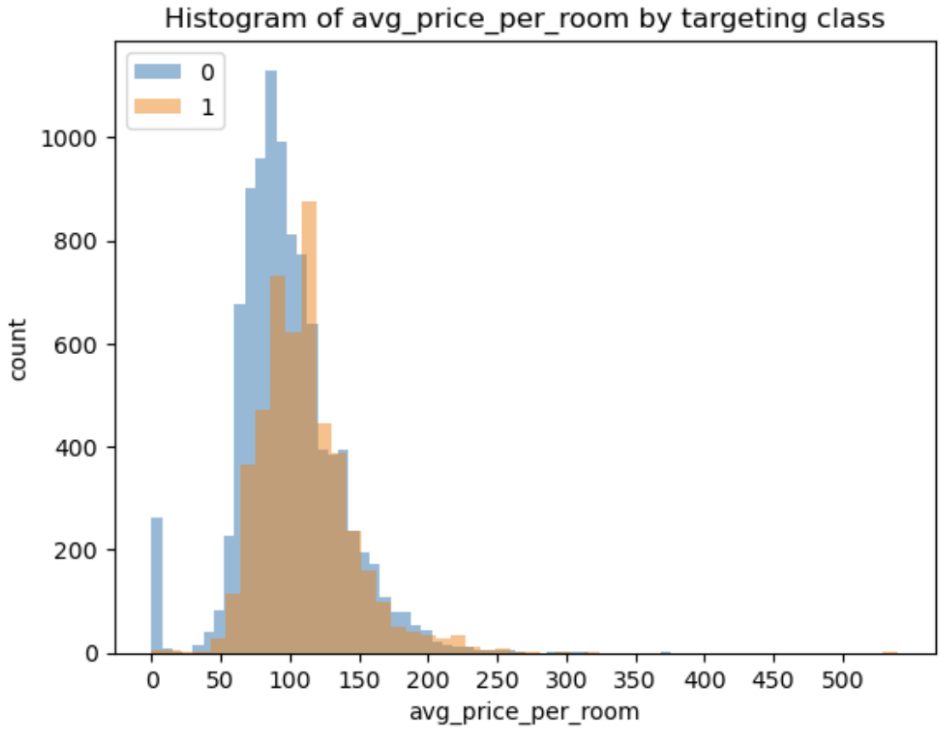
Visualizations

Next, we see some visualization to see how each feature is related to the target feature that is the booking status, we do that for **lead time** and the visualisation is as follows:



From the above we notice that the proportion of cancelled bookings is more when Number of days between the date of booking and the arrival date is greater than approximately 150 days hence this shows that there is some effect of lead time on the booking status.

We do the same for another feature which is average price per room by the booking status and we get the following visualization:



From the graph above we notice that the proportion of cancelled bookings is more when Average price per day of the reservation is greater than 110 euros.

These are just examples of some visualizations we can make with the features to better understand them.

Feature Engineering

We can also add some columns(features) to our dataset that would help us better understand the problem and the relationship between these features and the target class. We found certain features could be derived from the given set of features that would give us better insight into the booking and cancellation patterns of our customers.

We added the following features:
1) **Total Number of Bookings,** which tells us the total number of bookings a customer has made regardless of booking status.

2) **is_holiday_season** that lets us know if the booking was made in December or January which are usually holiday months of charismas. Adding such features would help the hotels be better prepared during holiday season.

Caveats associated with is_holiday_season feature:

We've only treated December and January as holiday months, however there can be other situations where a lot of bookings are made. E.g. Long weekend in April for Easter or Mid-term breaks for students. We did not explore such patterns which could have potentially been useful.

Getting data ready to fit models on:

We performed pre-processing on our features to analyse our data better.

- ➔ All the numeric features were scaled.
- ➔ Categorical features were encoded with One-Hot Encoder

Models Used

1) Linear Regression: to get a comparison point for the other models we fit on the data. This was an easy choice since we are dealing with a categorical result.

2) Decision tree Classifier

3) Random Forest Classifier

4) LGBM

We tried different models and Tree-based ensembles in an attempt to pick the one that performs best with our scoring metric (in our case: recall).

Deciding the scoring metric: In this course we learnt that accuracy is not always the best metric, we chose recall as our scoring metric because we want to identify cases where we wanted to minimize false negatives.

We chose to further optimize the LGBM classifier since that was the model that gave us the best recall.

Caveat: if we chose accuracy as our scoring metric, decision tree would have been an easy choice, however it was reporting an training accuracy of ~99.6% which is a clear case of our model memorizing our data, which is not ideal. Therefore, if we chose accuracy as our main scoring metric, we would have optimized the decision tree classifier to not overfit.

Final Result:

After optimizing our LGBM classifier, we ended with a train recall of 92.7% and a test recall of 71.6%, which means that if we put this model to use for our customer base, we would be able to identify 71.6% customers who are wrongly being classified as not cancelled but cancelled. Thus, we can prevent these customers from cancelling or be prepared to use their cancelled reservations for other/extra customers.