Business Intelligence

Assignment 3: Data Analytics

Martí Paulet López
TU Wien
Person B
e12305831@student.tuwien.ac.at

Álvaro Puig Bieger
TU Wien
Person A
e12305794@student.tuwien.ac.at

TASK 1: BUSINESS UNDERSTANDING

TASK 2: DATA UNDERSTANDING

2.1: Attribute types and their semantics

Our dataset consists of 31 attributes and 119390 rows, describing reservations made in two hotels in Portugal. Hence, each observation contains a lot of information about the details of the reservation. We will now present the attributes and their semantics:

- **Hotel**: Categorical variable with two categories, *City_hotel* or *Resort_hotel*. It denotes which of the two the reservation was made. The distribution is a bit imbalanced, since the *City_hotel* has 79330 observations, almost 2/3 of the entire dataset.

- **Is_canceled**: binary variable telling us if the reservation was cancelled (1) or not (0). As most reservations do not get cancelled, we also have almost 2/3 of the dataset with value 0 in this attribute.

- **Lead_time**: numerical value representing the number of days elapsed between the entering date of the booking into the PMS and the arrival date, hence the advance with which the reservation was made. Its values go from 0 to 737 days, and the mean is of 104 days. This can tell us information about the behavior of the clients.

- **Arrival_date_year, arrival_date_month, arrival_date_week_number, arrival_date_day_of_month**: these attributes are all pretty self-explanatory, they represent the year for which the reservation was made (the dataset contains information from 2015 to 2017), the month (January through December), the week number (1 to 53) and the day of the month (1 to 31). Even though they all are real numbers except the month, all of them are categorical attributes, since they represent a category inside the year, or the year itself.

- **stays_in_weekend_nights, stays_in_week_nights**: these two values represent the number of weekend or weeknights the reservation comprises. The sum of the two will give us the total number of nights in the reservation. Both are numerical integer values, with weekend nights going from 0 to 19, although 44% of the dataset is 0; and weeknights going from 0 to 50, where here only 6.4% of the data has value 0.

- **Adults, children, babies**: these are all three numerical values representing the number of adults, children, and babies in each reservation. Children and babies have a surprising number of 0's (children: 92.8%, babies: 99.2%), and adults are mostly reserving by pairs (75.1%). Furthermore, there are 4 observations with null values in the children attribute. There is a clear imbalance in these attributes throughout the data.

- **Meal**: the type of meal booked for the reservation, it can take 5 possible values. It is a categorical attribute, not very interesting.

- **Country**: represents the country of origin of the clients. Categories are represented in the ISO 3155–3:2013 format. This is a categorical variable containing 177 different categories. We can see that the most common nationality of guests is Portuguese, followed by the British and the French, with Spain fourth. There are 488 null values for country throughout the dataset.

- **Market_segment, distribution_channel**: market segment designation and booking distribution channel. These two categorical variables take each 8 and 5 different values, but they do not give useful information.

- **Is_repeated_guest**: binary variable that tells us if the guest has already been to the hotel before (1) or not (0). There is a huge imbalance in this attribute since 96.8% of the guests have never been before.

- **previous_cancellations, previous_bookings_not_canceled**: these variables represent the number of previous bookings that were cancelled by the customer prior to the current booking, and the number of previous bookings not cancelled by the customer prior to the current booking. They are both numeric variables. They are both extremely skewed and almost full of 0's.

- **reserved_room_type, assigned_room_type**: The reserved_room_type is the code of the room type reserved. Code is presented instead of designation for anonymity reasons. The assigned_room_type is the code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to overbooking or other situations. Both of these variables are categorical, with 10 possible values for the reserved room and 12 possible values for the assigned room. This difference could be explained by the hotels having special rooms not available for reserving, but being there in special situations for 'backup'. We can see that for both variables, the most common type of room is A, with quite a big difference. It is in both cases more than half of the observations.

- **booking_changes**: This variable represents the number of changes/amendments made to the booking from the moment the booking was entered on the PMS. It is a numerical variable, going from 0 (85%) to 21.

- **Deposit_type**: Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit, Non Refund, and Refundable. Once again, the vast majority of the observations have No Deposit, so in the data preparation we might need to delete or play with this variable to see if it is useful.

- **Agent**: ID of the travel agency that made the booking. Even though this variable is numerical (the values are numbers) it represents a categorical decision: each agent is represented by a number, thus each agent is a category. There are 333 different categories (agents) for this variable. The problem is there are a lot of missing values for this attribute (13.7%).

- **days_in_waiting_list**: Number of days the booking was in the waiting list before it was confirmed to the customer. It is therefore a numerical discrete variable. We can see that the vast majority of reservations are confirmed in the same day they're made, but there are some cases its not. The maximum a customer has had to wait has been 391 days, while the average is 2.3 days.

- **customer_type**: Type of booking, assuming one of four categories: Contract, Transient, Transient-Party and Group. The most common category is Transient, with 75% of the reservations.

- **Adr**: Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights. This is our objective variable, the one we want to predict using a regression model. It is a numerical variable. There are 8879 different daily rates recorded in the dataset, with the maximum being 5400 euros and the minimum being -6.38. In the data preparation phase, we will try to understand why there is a single negative value in this column.

- **required_car_parking_spaces**: Number of car parking spaces required by the customer. It is a numerical variable, ranging from 0 to 8. It is highly imbalanced (93.8% of it are 0's).

- **total_of_special_requests**: Number of special requests made by the customer (e.g. twin bed or high floor). Numerical variable ranging from 0 to 5, once again most values being 0 or 1.

- **reservation_status**: Reservation last status, assuming one of three categories: Canceled; Check-Out or No-Show. Most of the observations fall in the Check-Out category, followed by the Canceled, and very few are No-Show.

- **reservation_status_date**: Date at which the last status was set. It is a variable of type Date, which is not easy to work with. Most likely, we will erase it in order to be able to work better with our data. The day with the most status updates was the 21st of October of 2015.

2.2: Statistical properties

Now that we know a bit more about our dataset and what it contains, let's check some of its statistical properties. We've already talked about individual attributes and their most important statistics, so let's now focus on the correlation matrix of the data. Since a lot of our attributes are binary or categorical, we have computed three different matrices: the first one, only with numerical attributes; a second one also with binary attributes; and a third one adding a couple of variables using one-hot-encoding.

Numerical Data Correlation Matrix

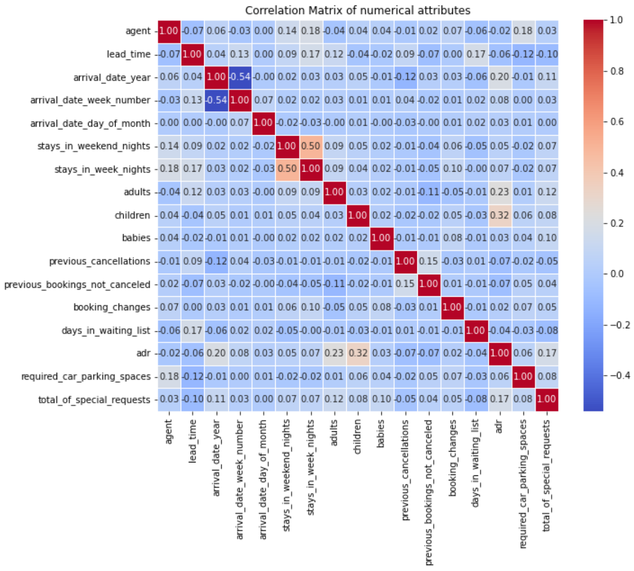

Figure 1: Correlation matrix of numerical attributes

We can see that there aren't a lot of highly correlated variables so far: arrival_date_week_number with arrival_year, for obvious reasons, week and weekend nights, and then we could say that children and adr are relatively correlated, with a value of 0.32.

Numerical and Binary Correlation Matrix

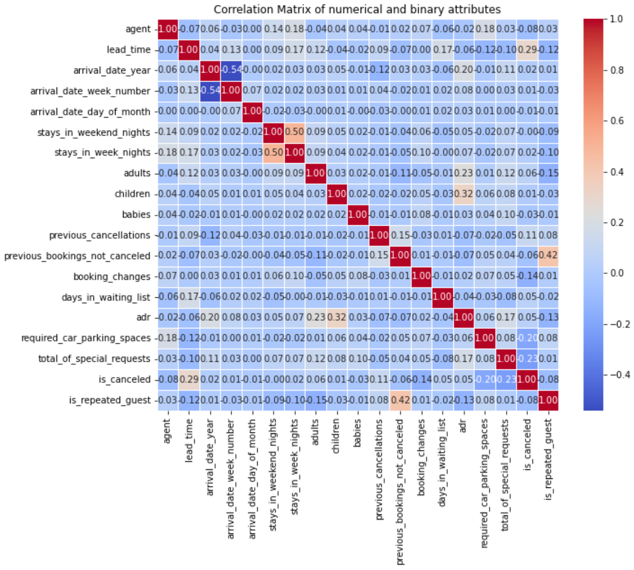

Figure 2: Correlation matrix of numerical and binary attributes

In this new matrix, we only have some new correlation between previous_bookings_not_canceled and is_repeated_guest, which makes a lot of sense given the data.

Numerical, Binary and Categorical Correlation Matrix

We cannot compute the correlation matrix for all the categorical variables using one-hot-encoding, since it would be a 1518x1518 matrix, and there would be too much (or none at all) information to extract.

However, there are certain categorical attributes that do seem quite important, and that do not have that many categories to choose from. Let's try and choose them, one-hot-encode them to see if we can get a decent sized matrix then. We will keep the attributes 'hotel' (to see if it is the city or the resort hotel), as well as 'reserved_room' and 'assigned_room'.

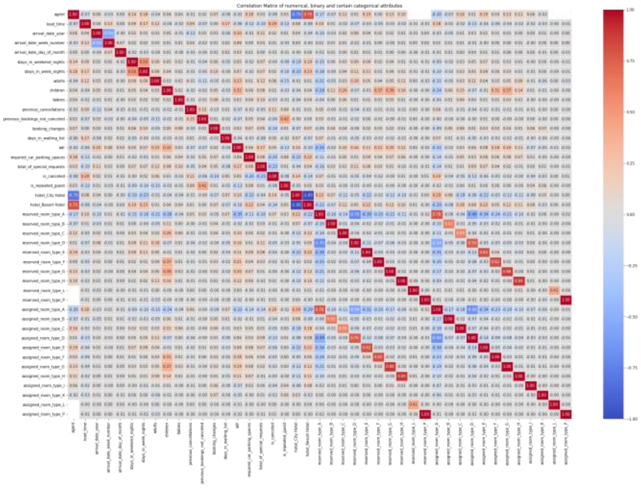

Figure 3: Correlation Matrix of numerical, binary and certain categorical attributes (check appendix for clearer view)

Now we have our correlation matrix of numeric and binary variables, as well as the 'hotel', 'assigned_room' and 'reserved_room' attributes using one-hot-encoding. As we could imagine, there is quite a high correlation between the assigned_room and reserved_room attributes, so it might be wise in the preprocessing steps to eliminate one of the two attributes.

2.3: Data Quality

Missing Values

We're now going to check the missing values of the dataset, although we already got that information on the profile report we did for part 2.1. We'll check the results and try to explain the reason they could be missing.

In total, there are 16832 missing values in the whole dataset. Doing some computation, we can find the missing values per column, and the percentage of observations they represent:

- Children: 4 – 0.00%
- Country: 488 - 0.41%
- Agent: 16340 – 13.69%

Firstly, checking the observations with the 'children' missing values, we can see that all four of them were cancelled reservations. Hence, we could think that the parents did not say if there were kids or not in the reservation, and the hotel could not verify it since they did not go through with the reservation. Since these are only 4 observations which correspond to less than 0.01% of the dataset, if needed we could erase them without it affecting our results.

In the 'country' null values, we believe it might be possible that the country of origin of those clients was not recognized by the country in which the hotels are (Portugal), which could explain why no value was inputted. Once again, these observations only amount to 0.41% of the data, which is almost negligible. If needed, we could erase them or input the top country (Portugal) to avoid losing more information.

Lastly, On the 'agent' attribute, there are 16340 observations with a null value, which corresponds to 13.7% of the dataset. Hence, we cannot eliminate all those observations. The agent attribute explains the ID of the travel agency that made the booking. We do not believe this attribute to be of much use, and furthermore, we can see that it is very highly correlated with the hotel attribute. Given this, we think the best idea is simply to drop this column and keep using the dataset without it, as there are too many null values to ignore them or fill them with the top Agent value. This field could be missing because some people do not use an agent, or this information was not required when making the reservation.

Data provenance and Data cleansing

The data comes from the article: 'Hotel booking demand datasets' from Nuno Antonio, Ana Almeida, Luis Nunes.

The data describes two datasets with information about hotel reservations. There are 2 types of hotels: a resort and a city hotel. The structure of the two datasets is identical, and every observation is a hotel reservation. Both databases include reservations scheduled to arrive between July 1, 2015, and August 31, 2017. Since this is real hotel data, all elements pertaining to customer identification were deleted.

Furthermore, the data was cleaned by Thomas Mock and Antoine Bichat as part of #TidyTuesday. The following code in GitHub explains how it was done: GitHub.

This is probably why it has so few null values, the data is complete and anonymized correctly, and all the attribute's values make sense.

Uneven distributions

Using the data profiling report from the beginning of the section, we can check that a lot of our attributes have a skewed distribution, due to the high number of zero values present in the data. However, this is not concerning for us, since those zero values make sense due to the nature of the attributes themselves.

Some examples are the attributes 'babies', 'children', 'required_car_parking_spaces', and others.

2.4: Visual Exploration of the Data

Now that we've explored our data, checked the correlations between the values and understand the attributes better, let's create some plots to try and see if there are any apparent relations or hypothesis we can formulate.

Relation between 'adr' and 'children', 'adults'.

One could guess that the price of a room might increase whenever the number of occupants increases, rather in a linearly fashion. Let's plot the scatterplots of 'adr' versus 'adults' and 'children' to check it out.

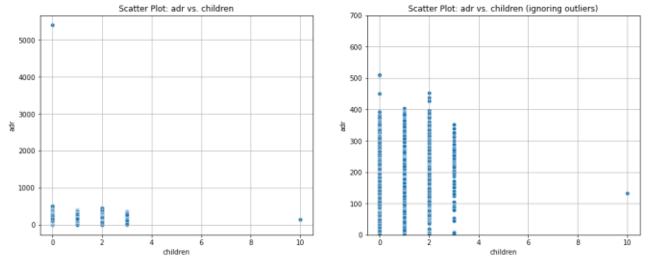

Figure 4: Scatterplot of ‘children’ and ‘adr’.

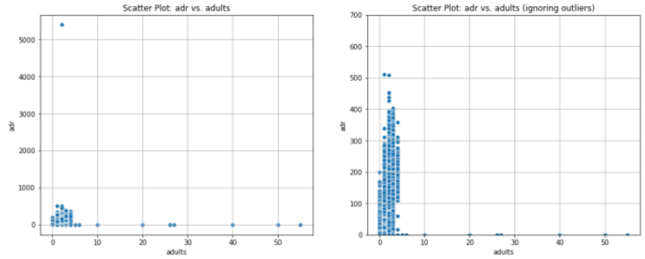

Figure 5: Scatterplot of ‘adults and ‘adr’.

We can see there is a clear outlier in the 'adr' attribute, where the price is around 5400. In order to better see the relation, we have plotted the scatterplots with the outlier, and then set the x and y limits to ignore it.

After consideration of the plots, there does not seem to be an obvious linear relation between the average daily rate and the number of adults and children in the reservation.

Difference in the average 'adr' for the 2 types of hotels

Now, let's check if there is a clear difference between the average daily rate depending on the hotel we're in. This way, we could draw some conclusions to know if we will need to separate the dataset for our future regression model (if the 'adr' is too different, it might not make sense to make only one model when it should be two).

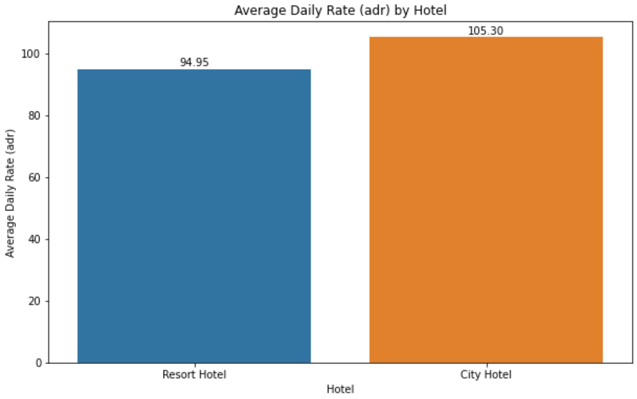

Figure 6: Bar plot of ‘hotel and ‘adr’.

We can see that the City Hotel is on average 10.35 euros more expensive than the resort hotel. This is not a massive difference, so we can be more assured of the integrity of the dataset.

Relation between 'adr' and 'month_of_arrival'

As both hotels are in Portugal, which receives a very high number of tourists in the months of summer, let's check how the 'adr' evolves during the year, and if it reflects those fluctuations in the number of tourists.

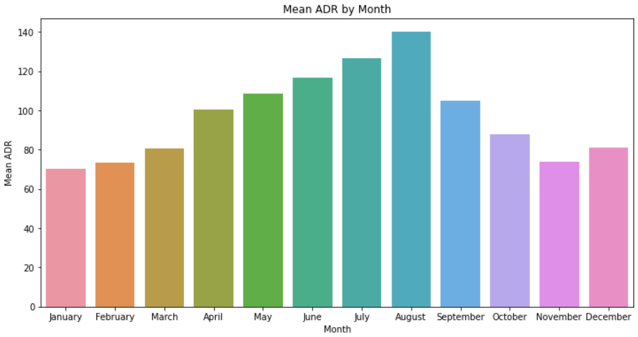

Figure 7: Bar plot of ‘arrival_date_month’ and mean ‘adr’.

We can see that indeed, during the months of June, July, and August, the 'adr' is at its highest of the year. We can also see a clear rise of the average daily rate as we approach those summer months.

Relation between 'adr' and ‘arrival_date_year’

Here, we simply want to check how the prices have evolved throughout the years present in the dataset. We want to check if staying in those hotels has been cheaper, equal, or more extensive over the years.

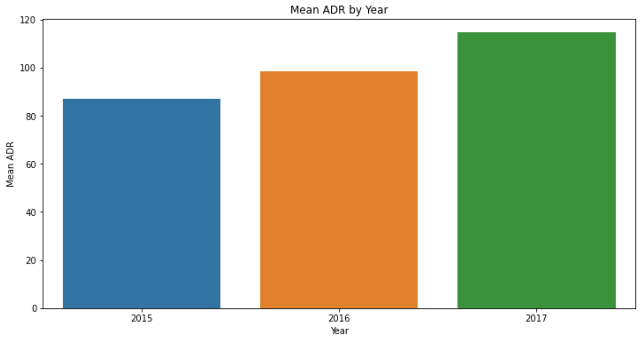

Figure 8: Bar plot of ‘arrival_date_year’ and mean ‘adr’.

We can see a pretty clear increase in the adr over the years: this could be caused by inflation, the economy of the country, or simply the hotel's strategies.

Number of observations by arrival year

Now, let's take a look at the number of observations we have for each of the three years in the dataset.

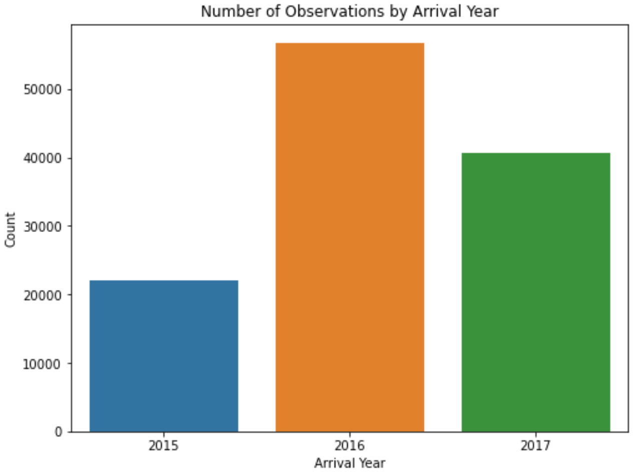

Figure 9: Count plot of ‘arrival_date_year’.

We can clearly see that we have the most information from reservations made in 2016. This is because the dataset does not include the whole 2015 and 2017 years, but it does include the whole 2016. Hence, it is fairly normal to get this plot. Ideally, we would have a similar number of observations from each year in order to have a more balanced dataset, but we do not think this will have much impact on our model.

'adr' trends over time throughout the years

Finally, let's use a line plot to check the evolution of the 'adr' during the year, per week.

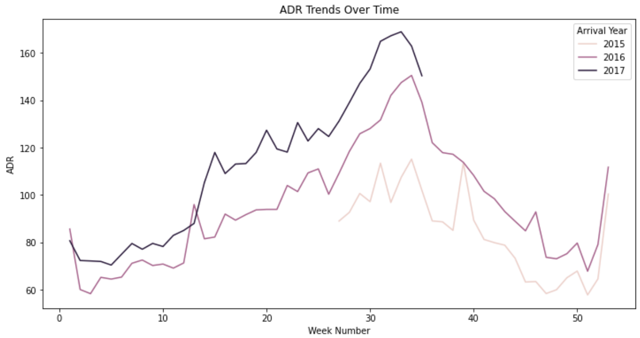

Figure 10: Line plot of ‘arrival_date_week_number’ and ‘adr’.

During all three years we can see a clear peak in the summer months, around the weeks 30 to 40. This is similar to the results we got in a previous plot.

2.5: Ethically sensitive Data

Taking a look at the data, the way it has been cleaned and anonymized, there is really no ethically sensitive data to extract. One cannot know the race, sex, or any detail about any of the clients of neither hotel. The only possibility of sensitive data would be the attributes 'children' and 'babies', which do tell you something about the clients and possibly their familiar status. However, I do not think this is a big issue, since all the data is anonymized and impossible to track back to the families.

There are, however, some unbalanced distributions inside the dataset, as we said before, that do need to be at least acknowledged, in order to fully understand what we're working with.

We have to say that the number of observations from the City hotel is 79330 (66.4%) whereas the Resort Hotel only has 40060 observations (33.6%). We doubt this will affect our results negatively, but we do have to know that we have more information from the city than from the resort.

Secondly, on the country attribute, the most common country is Portugal, which makes sense since it is where the hotels are. However, it quadruples the second most common country of origin, so we may need to be cautious if using that variable for our model.

2.6: Potential risks and additional types of bias

As explained before, we do not think there is a real risk by using this dataset, thanks to the cleaning and preprocessing work of its authors.

We would get an external expert to consult on the 'children' and 'babies' variables, to see if there is any possibility of the families being tracked and any personal information becoming public. We also think that the 'country' attribute could be a problem, but we doubt it will have weight in the regression, since the hotels cannot discriminate by country of origin to give prices for their rooms. So, if it is the case that this attribute is biased, it would then mean the hotels are doing something illegal and discriminating against their clients based on their nationality.

Finally, as we said before, there is a big imbalance between the hotels in the dataset, but we do not think this will have a bias in our results.

2.7: Actions for Data Preparation

There are several actions needed for the Data Preparation phase in order to start working with our dataset.

Firstly, we have to say that we already started this Data Preparation at the beginning of the notebook, by eliminating the column 'Company'. This column represented the ID of the company/entity that made the booking or was responsible for paying the booking. As it was filled at 95% with Nan values, we just thought it best to drop it at the beginning as it was giving us no useful information.

Secondly, as stated before in the 'Missing values' part, we are going to drop the 'agent' column. It has too many missing values, we believe it might give us problems, and it is highly correlated with the hotel variables which could cause issues in the regression.

Thirdly, we will erase the observations that have null values in the 'children' column. They correspond to less than 0.01% of the data, and we believe that the 'children' attribute may be important in the regression model. We do not want problems with the code and this way we are assured of that.

Afterwards, we will drop the column 'arrival_date_year', as it is highly correlated with 'arrival_date_week_number', and thus it is not of much use.

Then, we will eliminate the outliers in the 'adr' column: there are two observations that can cause us problems. The first one has a negative adr, which is impossible because it would mean the hotel had to pay the client for him to spend the night. This does not make sense, so we believe there was a mistake, and the easiest thing to do is simply to delete the observation. The second observation that might cause problems is the outlier which has an 'adr' value of 5400, which makes no sense in relation to the rest of the data. We also believe that there was a mistake in this reservation, and the adr value is erroneous. Hence, for it not to mess with our regression model, we will eliminate it.

We will also drop the column 'assigned_room_type' since it is very highly correlated with 'reserved_room_type', and we do not want useless information.

And finally, we will drop all columns that are not numerical or binary attributes, excepting 'hotel' and 'reserved_room_type'. For those two attributes, we will use one-hot-encoding in order to turn them into a set of binary attributes that represent those categories.

After careful consideration, we believe that having 31 attributes is too much for our regression model. Furthermore, most of the categorical variables have too many categories to be able to do one-hot-encoding and work with them, and we believe that most of them are practically useless when trying to predict 'adr'.

We also thought of using over or under-sampling in order to create a more balanced dataset in relation to the 'hotel' attribute, but after thinking it through, we have decided against it. We do not want to lose information while under-sampling nor create 'recycled' observations through over-sampling, and we do not think the data is too imbalanced for it to mess with our models.

However, what we do want to do is separate the data into two subsets: the first subset will be the complete dataset, all together. The second subset will have the data separated by their 'hotel' value: this way, we will create a regression to predict 'adr' for both hotels combined, and also for each hotel separately. We will do this in order to check if the model differs a lot between the hotels, or if on the other hand, the type of hotel is not very important to decide the 'adr'.

TASK 3: DATA PREPARATION REPORT

3.1: Data Preprocessing

In this section, we will simply code the preprocessing steps we mentioned in the previous section 2.7. The code will be available in the Jupyter Notebook (Section 3.1). However, here is a quick recap to what we will do to the dataframe:

- Dropping ‘company’, ‘agent’, ‘arrival_date_year’, ‘assigned_room_type’ and all other non-numerical or binary columns except ‘hotel’ and ‘reserved_room_type’.
- Dropping four observations with null values in ‘children’, and two outlier observations for ‘adr’ (one negative value, one too large).
- One-hot-encoding 'hotel' and 'reserved_room_type' variables.
- Create the two working subsets for the future model.

After these steps, we end up with three interesting datasets: *data_final*, our dataset containing information about both hotels but only the columns that interest us, *data_city* containing only the information on the city hotel, and *data_resort*, containing only the information on the resort hotel.

3.2: Derived Attributes

After some consideration, we've managed to find some derived attributes that might be helpful for our business objectives.

Firstly, we've already done it, but one-hot-encoding our categorical variables 'hotel' and 'reserved_room_type', in order to get binary attributes that can be used in our regression model, is the first step in our derived attributes.

Secondly, we will create a new attribute for each reservation, called 'total_guests'. This will simply be the sum for 'adults', 'children' and 'babies' for each observation. This way, we will see if the category of each person is important, or if simply the total number of people has a greater effect on the 'adr'. The only problem we see with this is the high correlation this attribute will have with the three it is composed of.

Then, we will also create the attribute 'total_nights_stay', which will simply be the sum of 'stays_in_weekend_nights' and 'stays_in_week_nights'. Once again, this will be helpful in determining if there is a difference in staying during the weekend, or if the 'adr' is simply determined by the total number of nights.

We've also considered the use of binning for the 'lead_time' attribute, in order to create three categories: short-term, mid-term and long-term reservation. However, we believe this new way of looking at the lead_time attribute could be helpful if we were trying to study the clients' vacation habits, but we do not see the utility for our business objectives. Furthermore, it would only cause problems in our regression model.

Furthermore, we thought about creating a binary indicator for the 'booking_changes' variable, instead of it being a numerical attribute. However, after more consideration, we reached the conclusion that we cannot know if the actual number of changes is important, and thus we do not want to lose information by changing it to a binary variable with only 0 or 1 as possible values.

Finally, we also will add a 'family' attribute, which will simply be a binary variable with value 1 if the reservation has 'children' and / or 'babies' different than 0. We think this might be helpful to quickly recognize if the reservation belongs to a family vacation and study the 'adr' based on that.

The last derived attribute we thought of is 'booking_window': calculate the difference between 'lead_time' and 'days_in_waiting_list' to represent the effective booking window of each reservation. However, we thought this could be interesting if we wanted to study the client's habits, but it does not really align with our use of the data in this case.

We will proceed with the changes in section 3.4, along with describing all the preprocessing steps made to the data.

3.3: External Data Sources

Given how complete our dataset is, we find it hard to find external data that could be useful for us to better predict 'adr'. However, there is some information that could be of use.

Firstly, we believe having surveys of clients after they've checked out, rating their stay from 1 to 10, could be useful in order to better understand how satisfied they were with the services. This could give us a better understanding of the hotel, if people will or will not come back, etc.

Secondly, we could insert into the dataset a local events calendar, to check if the reservation coincides with a local holiday, big conference in the area or any festivals. This could be useful to understand the fluctuations of the 'adr', since normally these events attract more people, and that raises the prices.

Lastly, having a calendar of local Economic Development Plans could also help. By exploring local economic development plans or projects that might attract visitors and impact hotel demand, we could have a better shot at predicting the 'adr', since as we said before, that could be a reason to increase prices for hotels.

After some consideration, we won't be applying any of this external data, since we do not think the complications it would bring would be worth it for the prediction model.

3.4: All pre-processing steps

Finally, we're going to recap all pre-processing steps applied to the data:

- Dropping the columns 'company', 'agent', 'arrival_date_year', 'assigned_room_type', as well as all other categorical attributes except 'hotel' and 'reserved_room_type'.
- Deleting observations with 'children' values as null and the two 'adr' outliers.
- One-hot-encoding the 'hotel' and 'reserved_room_type' variables to transform them into binary attributes.
- Creating the two subsets for the future regression model: one with the whole dataset, and then two dataframes containing information from each hotel.
- Now, we're going to create the following attributes: 'total_guests', 'total_nights_stay' and 'family', which will be created following the directions given in section 3.2.

We do not need to scale our data given its nature, and thanks to the cleaning and preprocessing work done by the authors of the dataset, this is everything we need to do to prepare the data to create our regression model.

The code for the creation of 'total_guests', 'total_nights_stay' and 'family' attributes will be in the Jupyter Notebook in Section 3.4.