**Data Science Project Protocol  
Naama Jan**

**Introduction:**

For this project, I decided to choose the Airbnb Berlin dataset.  The dataset can be found on Kaggle:  
<https://www.kaggle.com/brittabettendorf/berlin-airbnb-data>

[Airbnb](https://www.airbnb.com/) is an online marketplace where people can offer and rent out their houses and apartments (properties). Airbnb became extremely successful for tourists and business people, who are looking for an overnight accommodation.

The datasets are based on properties list with description (features) and information whether the properties are being rented in each day from November 2018 until November 2019.

The project goal is predicting the probability of the properties to be rented in the next 3 months. Defining the outcome variable as booked if there are at least 70 rented days in the 3 months following every month.

The outcome variable is the "booked up" state of the following 3 months for each month starting in March until August (the first prediction starts in March since I'm using November-February for training. August is the last month since the last prediction is from August until November). I define the booked-up variable as a binary variable where it's set to 1 if at least 70 out of 90 days are booked and 0 otherwise.

I plan to examine the occupancy of the properties, according to the variables in the model. In general, there are variables that I expect to have high significant influence on the outcome. For example, price, weather, location (neighborhood), the distance to the city center, the size of the property, room type, reviews (especially review scores rating) and etc.

**Data**

The main datasets are based on properties list with description (features) and information whether a property is being rented in each day from November 2018 until November 2019 (every record describes property in a day).

As mentioned in the introduction, our target is predicting whether a property will be rented in the next 3 months (defining the booked\_up\_target base on occupancy of the property).

These datasets are taken from [Kaggle](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data) and ensembled from 6 different csv files:

* [Calendar Summary](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=calendar_summary.csv)
* [Listings Detailed](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=listings.csv)
* [Listings Summary](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=listings_summary.csv)
* [Neighborhoods](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=neighbourhoods.csv)
* [Reviews Detailed](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=reviews.csv)
* [Reviews Summary](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=reviews_summary.csv)

For enrichening the main data sources, I used [climatestotravel](https://www.climatestotravel.com/climate/germany/berlin) website for getting average weather data. Assuming that weather info can influence on tourism in Berlin which can directly influence the demand for properties.

Importing the following datasets from this website:

* Sunshine hours
* Average precipitation
* Average temperatures

The data sections in this project are divided to the following steps:

1. The first step in the project is collecting the datasets from Kaggle to SQL server and creating a flat table (by relevant joins between the tables). Most of the features are created at this stage.
2. There are enrich features that requires analysis of text. These feature have been added to flat table using “1- Addition to Flat file.ipynb” notebook.
3. In “[2 - Berlin\_Airbnb\_EDA.ipynb](https://github.com/NaamaJan/airbnb-berlin-availability/blob/main/2%20-%20Berlin_Airbnb_EDA.ipynb)” notebook, I perform an analysis of the detailed Berlin listings data. I am using visualizing and analyzing data to extract insights from the variables in the data.

In the beginning, I used descriptive statistics to explore the data, which can help describe the data set's basic features and obtain a summary of the data. Then, I perform Data Visualization analysis to provide an accessible way to see and understand trends, outliers, relationships, variability, patterns in data and to notice if there is a problem with data quality.

I have been paying attention to the correlations and the differences between the variables in the data set and describe the target variable "booked\_up\_target", his distribution and its relationships with the variables.

This section also helped me create new variables or perform variable transformations.

1. [3- Berlin\_Airbnb\_Data\_Cleansing.ipynb](https://github.com/NaamaJan/airbnb-berlin-availability/blob/main/3-%20Berlin_Airbnb_Data_Cleansing.ipynb) notebook includes reduce the bias by filling NA, fixing outliers and etc. This is based on the results and conclusions of the EDA section.

The methodology of handling outliers:

1. If the outliers are a mistaken observation - replacing the outliers with na.
2. If the outliers generate any false correlation with the outcome - replacing the outliers with na.
3. If the removal of the outliers changes the distribution of the outcome but not the correlation - replacing the outliers with na.
4. If the removal of the outliers changes the both correlation and the distribution of the outcome - we can't replace the outliers with na. In this case we will need to perform a data transformation or make the variable categorical and thus divide it into groups that one of which will be "missing" or use models to predict the missing variables.

The methodology of handling missing values depends on the mechanism of missing generation for each variable. For each variable we need to decide the correct method to be applied:

1. Dropping rows with more than 60% outliers.
2. checking the number of NA in columns:
   1. If the percentages of na are greater than 70% -> dropping the column.
   2. If the percentages of na between 40%-70% -> transforming the variable categorical.
   3. If the percentages of na are lower than 40% -> depends on the mechanism of missing generation for each variable. For each variable, we have to decide the correct method to be applied. If the mechanism of missing is MNAR -> There is an explanation of why the value is missing. In this case, we can transform the variable categorical or drop the column, we cannot do imputation. Otherwise, if the missing mechanism is MCAR or MAR, we can use imputation techniques.
3. [4- Feature Enrichment.ipynb](http://localhost:8888/notebooks/Documents/projects/airbnb-berlin-availability/4-%20Feature%20Enrichment.ipynb) notebook includes 3 way of adding features:
   1. Feature Extraction: obtaining new features from existing features.
   2. Feature Engineering: transformation of raw data into features suitable for modeling.
   3. Feature Transformation: transformation of data to improve the accuracy of the algorithm.

## For creating [Data retrieval protocol](https://github.com/NaamaJan/airbnb-berlin-availability/blob/main/Data%20Retrieval%20Protocol.xlsx) I used [7 - Data Retrieval Protocol Helper.ipynb](https://github.com/NaamaJan/airbnb-berlin-availability/blob/main/7%20-%20Data%20Retrieval%20Protocol%20Helper.ipynb) **notebook.**

**Models**

In the model section I am planning to divide the data to train, dev and test datasets partitions.

The datasets should be spit only after shuffle and balance between the 3 datasets.

pyMechkar package include utils for splitting the data, shuffle and supply indication whether the data is balanced (tuning with seed and prop parameters). The flat table includes ~157K records, so it seems reasonable splitting having propitiation of ~60%, ~20% and 20%. I will use this package.

The first stage of this splitting is defining 20% of the records as test partition, keeping it aside and not using it until the end of the project.

The 80% leftovers should be split to train and dev with proportion of 80%-20%. The train dataset will be used to train the models while the development dataset will be used for assessment of the model performance.

The outcome 'booked\_up\_target' is binary variable which is set to 1 if at least 70 out of 90 days are booked and 0 otherwise. For training the model I will use wide range of classifier models with default params, selecting the model which perform best on the train and dev partition (high score and balance scores between 2 partitions).

The score will be determined by Area Under the Curve (AUC) metric, which can be useful in the scenario which the 72% of that target are “1” and 28% are “0”.

Once selecting the best model (base on train and dev), I will make fine-tune of the hyper-paremeters:

1. Depending on the model, create vectors with a wide range of different values for part of the parameters that may affect the performance of the model.
2. Using Grid-Search for finding the best matching parameters.

**Deployment of your model**

As mentioned above this project goal is predicting in the end of every month whether a property will be booked or not in the next 3 months. This requires that in [Calendar Summary](https://www.kaggle.com/brittabettendorf/berlin-airbnb-data?select=calendar_summary.csv), there will be an history of at least 3 months (listing id, date and is booked) and all raw features in the other 3 tables.

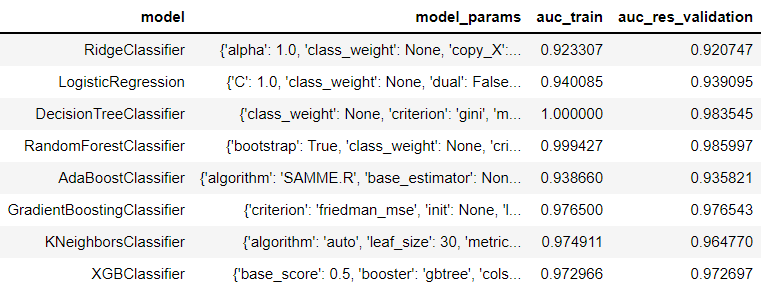
After finished all sections: EDA, Cleansing, Enrichment and feature selection I got selected features which don’t include listing\_id. I added listing\_id to the list of features because it will be useful filter in the model prediction.

The prediction models that I run are Ridge, Logistic Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, KNN and XGB.

Base on AUC performance measurement selecting XGB as the classifier model for this project. Almost all models did well, but my selection of the model based on high score of the train and dev with minimum delta between the 2 (low chance of overfitting).

Gradient Boosting could have been also good candidate.

The results of the models:



As mentioned, I selected XGB and I tuned the parameters from the default and got the “selected” model object.

This model object can be persist with pickle/pkl or pmml for future use so I don’t need to re-train it (pmml is useful if we would like use other languages than python to read this file and make prediction on the model – for example if we would like make real time prediction with Java).

I will need to create a script (ETL) that I would run at the end of every month. The script should use all raw tables to build flat table of listing id to the 55 selected features. I should re-use methods that I have created in data cleansing and enrichment for creating those features.

I will load the model from disk (or any other storage) to model object and use predict method predicting booked\_up\_tarket.

I will add const columns target\_start\_date\_period (current date) and target\_end\_date\_period (current date + 3 months) to dataset.

All rows in the same prediction date get the same values in these 2 columns.

This new dataset will be dump to a csv or append relational table.

We can query this table filtering listing id, target\_start\_date\_period and target\_end\_date\_period (these results can be sent to the property owner).

We could also use this table to make an assumption to new properties if we can fill up the 55 features.

This project will build in layers and requires few levels of testing:

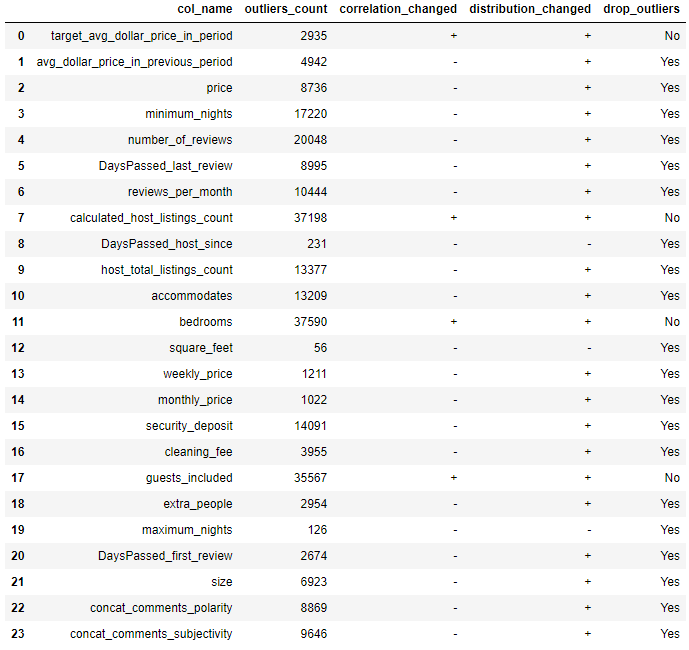
1. Collecting the raw data. Validation of the raw data should be done by BI or any owner of this table. Also, many times the users of the tables validate the inputs. We should have metrics if we get new values of features that we have not seen before (especially for categorical features).
2. Creating flat table in SQL requires some validation by me.
3. EDA, Data cleansing, feature enrichment and features selection requires unit testing for the functions (which I add). Also, once in a while if the performance of the prediction changed, I need to make update and fixes of the above.
4. Before the train of the model, I split the dataset to train, dev and test and validate the data between partitions is balanced.
5. I need to have a list of expected values for each feature, I need to decide for each feature which value to set in case of new value or na.
6. Except of checking the performance of the test partition dataset in the “lab” (happened only in the train), I need to check in production the performance of the model prediction (the etl prediction). Aka as mentioned above keeping records of the prediction by listing id and dates and compare them with what happened in real. I need define dashboard with metric that compare what happened in real with my prediction. If there are any dropout I need to start investigate back.

**Results**

The final datasets we use in all the section of the model creation is distrusted as follow:

|  |  |
| --- | --- |
| Dataset partition | Number of rows |
| Total | 157,864 (100%) |
| Train partition | 101,032 (64%) |
| Dev partition | 25,259 (16%) |
| Test partition | 31,573 (20%) |

As appear in “3- Berlin\_Airbnb\_Data\_Cleansing” I got outliers in the following columns:

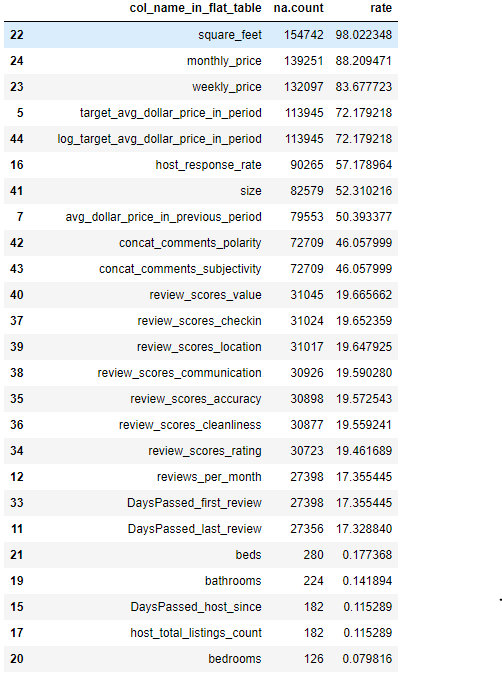


The methodology of handling outliers is mentioned in “Data” section in this document.

We can see above that for 20 features there was no change both in correlation and in distribution (only one or none), so I could replace them with “NA” (“Yes” value).

The other 4 columns (“No” value) there was a change in both correlation and distribution so I needed to use transformation for them (using log, sqrt and sigmoid).

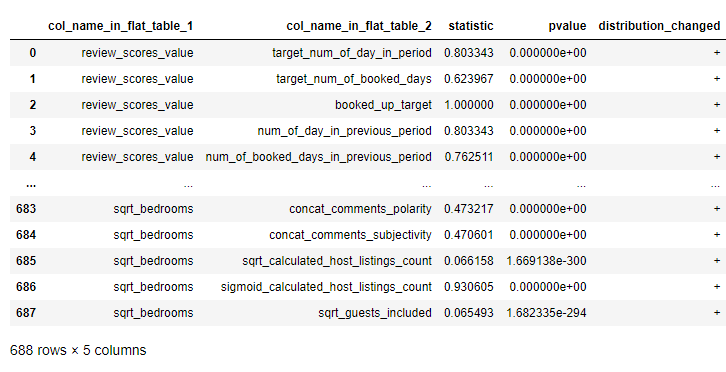
As appear in “3- Berlin\_Airbnb\_Data\_Cleansing” I got missing values in the following columns:



Handling missing values depends on the percentage of missing values in each column:

1. “square\_feet”, “monthly\_price”, “weekly\_price”, “target\_avg\_dollar\_price\_in\_period” and “log\_target\_avg\_dollar\_price\_in\_period” had more than 70% missing values so I dropped this columns.
2. “host\_response\_rate”, “size”, “avg\_dollar\_price\_in\_previous\_period”, “concat\_comments\_polarity“ and “concat\_comments\_subjectivity” had between 40%-70% missing values so I transformed to categorical variable.

“review\_scores\_value”, “review\_scores\_checkin”, “review\_scores\_location”, “review\_scores\_communication”, “review\_scores\_accuracy”, “review\_scores\_cleanliness”, “review\_scores\_rating”, “reviews\_per\_month”, “DaysPassed\_first\_review”, “DaysPassed\_last\_review”, “beds”, “bathrooms”, “DaysPassed\_host\_since”, “host\_total\_listings\_count”, “bedrooms” and “sqrt\_bedrooms” had less the 40% missing values. For this case (less than 40% missing values) I used transformation to categorical because as mentioned in the notebook distribution\_changed was + for all these list and columns (I have not needed to use imputation).



* The final amount of data used (total, train, test, etc)
* The amount of outliers and the way of treating them,
* The amount of missing values and the methods used for imputing them,
* The distribution of the data (timeframes)
* חילקנו את הDATA
* ל3 חודשים TRAIN
* וחודש TEST
* יצרנו מדרג
* The methods used to transform the data and to generate new features.