**Hotel Sales Prediction:**

Objective: To predict sales for each given hotel on each future dates given the booking and search data for each hotel.

The following .py files which is used to evaluate the sales of the hotels along with description has been mentioned below:

1. *init\_probe.py* : This file has been used for initial probe of the data.
2. *dataset\_creator\_function.py*: This file contains all the functions that are used for creation of datasets for training model
3. *dataset\_creation.py*: This file is used to create a dataset which will be used to run models
4. *model\_training.py*: This file contain model fit function and evaluation function for fitting a model to the training dataset created
5. *prediction\_file.py*: This file is used to create dataset from validation data and predict using the fitted model

**init\_probe.py**

Functions present:

*date\_parsing\_fn:* Used to parse dates in the given data

*dates\_explorer\_displayer:* function used to print the min/max dates in the data for both booking and check-in

*data\_subset\_fn:* function to subset the data based on the given parameters – checkin date, city and hotel

Main:

* Checking summary of the data, the data types present and the min & max dates present.
* Checking whether number of unique hotels present is equal to the number of unique combination of city and hotel
* Checking unique values of attribute ‘flavour’
* Extracting data to write booking data(data\_txn) for one single hotel for easier understanding of data in excel
* Checking the min/max date for the validation data

**dataset\_creator\_function.py**

Functions present:

*txn\_data\_aggregator:*

parameters :- booking data, list of hotels

return :- A dataframe with aggregated sales of rooms for each hotel on each date

Code:

For each hotel from the list of hotels the data is created and appended to the resulting dataframe which will be returned. Booking data is subsetted for a hotel id, for this hotel id the max and min checkin date are calculated. Then all the dates are created that are present in between the min and max date. Now for each of these dates the number of rooms that are booked is calculated.

If for any entry for a given hotel the checkin date is less and checkout date is more than the date for which number of rooms is being calculated then it is added to the counter.

The column names in the final returned dataframe are Dates, hotelid, cityid, rooms\_booked

*demand\_data\_aggregator:*

parameters: booking data aggregated as created by above function (*txn\_data\_aggregator*), search data, flag suggesting whether its training or prediction

Return: None

The function calculates hotel specific and city specific aggregated values for each date present in the aggregated booking data.

Two pickle files has been dumped, one for values containing aggregated values on Date and Hotel id and the other containing aggregated values on Date and city.

These files will be used for creation of dataset in the function (*txn\_demand\_data\_compiler*).

*txn\_demand\_data\_compiler:*

parameters: booking data aggregated, city\_df (dataframe containing aggregated values on Date and city level as dumped by *demand\_data\_aggregator* function), hotel\_df (dataframe containing aggregated values on Date and hotel level), flag suggesting whether its training or prediction

Return None

The function writes a csv file which will be used model in case the flag is training and if the flag is prediction then it will used for prediction

The function merges the aggregated booking data and the files that contain data on aggregated city and hotel level, the data has been merged on ‘Dates’ and ‘hotelid’/ ‘cityid’

Variables like day of the week , max number of rooms and average number of rooms booked are also calculated. Finally dummy variable is created for cityid. The final file is saved as a csv under the name ‘final\_data.csv’ in case of training or ‘final\_data\_pred.csv’ in case of prediction.

**dataset\_creation.py**

This python file is run to create the dataset which will be used by model\_training.py to train a model. The file reads both the booking data and search data. From booking data list of all the unique hotels has been created which is used as a parameter for the function *txn\_data\_aggregator.* This function is called to return aggregated booking data dataframe.

Then *demand\_data\_aggregator* function is called to dump files as explained in the dataset\_creator\_function.py

The aggregated city and hotel dataframes are loaded and then *txn\_demand\_data\_compiler* function is called to create final data and save it into csv

**model\_training.py**

Functions present:

*model\_fit:*

parameters: algorithm (any algorithm like linear regression, gbm etc.), X (independent varaibles dataset), Y (dependent variable), evaluation\_fn (function defining accuracy metrics), performCV (True if cross validation is required)

Return None

The function splits the data into train and test data. The given model is fitted and the score is calculated using evaluation function. The scores are also printed.

K-fold cross validation is also run on the dataset if the value for performCV is True

*model\_evaluation:*

parameters: y\_true (actual values that are being predicted), y\_pred (predicted values)

return score (a number or percentage which is used to measure the performance of the model)

Main:

Dataset as created in dataset\_creation.py is loaded. Dependent and independent data is separated and algorithm to be used is defined. The *model\_fit* function is called with parameters described parameter.

The model is dumped as a pickle file which will be used in prediction.

**prediction\_file.py**

This python file creates the dataset from the validation data and the file is written as csv ‘final\_data\_pred.csv’.

This file is loaded again and prediction is done using the model as dumped in the model\_training.py. The result is written to csv as ‘result.csv’

Both the booking and search validation data is read from a csv.

Booking data is aggregated and then aggregated values for city/hotel and date level is calculated and finally combined in the same manner as done in dataset\_creation.py

final\_data\_pred which is created by the above is set of code is read again. Same set of dummy variable columns are created as present in the dataset used for training. The model is used to predict and the resulting file is written to csv containing both actual and predicted values.

***Note****: The order of file to run the prediction is as below:*

1. dataset\_creation.py
2. model\_training.py
3. prediction\_file.py