CS-236 Group Project Report, Phase I

Group number : 15
Fall 2025

Introduction:

This report deals with the phase 1 of the class group project, done via using **Pyspark.**The two datasets used here are **"customer_reservation.csv"** and **"hotel_booking.csv".**
The report has following section:

(A) Installing Docker and Pyspark in our local systems.
(B) Exploratory Data Analysis on the datasets
(C) Creating a Merged Data set for Phase II.

Installing Docker and Pyspark in our local systems:

- **Bufan** : The installation of PySpark is quite convenient. Firstly, creating a conda environment *"conda create -n pyspark"*. Then, *"conda activate pyspark"* to activate and execute *"pip3 install pyspark"*. By doing the steps above, Pyspark is ready for EDA. Pyspark offers lots of useful functions in pyspark.sql, which helps a lot when analyzing.
- **Sayantika** : I installed Docker (according to the instructions provided) and created a complete Apache Spark cluster development environment using VS Code DevContainers. This setup provides a fully configured multi-node Spark cluster with master, workers, history server, and comprehensive monitoring capabilities running in Docker containers with zero-configuration setup. I integrated the Jupyter notebook with it too, to deploy and deliver the projects aesthetically. I took help from a .github repository Github_devcontainer . I also integrated PostgreSQL into this devcontainer so that we can deploy the phase 2 and 3 of the project without any further setup.

Exploratory Data Analysis:

1. **Initial Data Analysis:** We started a spark session and imported the two datasets. The following table shows the findings:

Table 1:

Data set name	**num_Rows (Data points)**	**num_Column (Features)**
customer_reservation	36275	10

hotel_booking	78703	13

Schemas:

```
customer_reservation Schema
root
 ├── Booking_ID: string (nullable = true)
 ├── stays_in_weekend_nights: integer (nullable = true)
 ├── stays_in_week_nights: integer (nullable = true)
 ├── lead_time: integer (nullable = true)
 ├── arrival_year: integer (nullable = true)
 ├── arrival_month: integer (nullable = true)
 ├── arrival_date: integer (nullable = true)
 ├── market_segment_type: string (nullable = true)
 ├── avg_price_per_room: double (nullable = true)
 ├── booking_status: string (nullable = true)

hotel_booking Schema
root
 ├── hotel: string (nullable = true)
 ├── booking_status: integer (nullable = true)
 ├── lead_time: integer (nullable = true)
 ├── arrival_year: integer (nullable = true)
 ├── arrival_month: string (nullable = true)
 ├── arrival_date_week_number: integer (nullable = true)
 ├── arrival_date_day_of_month: integer (nullable = true)
 ├── stays_in_weekend_nights: integer (nullable = true)
 ├── stays_in_week_nights: integer (nullable = true)
 ├── market_segment_type: string (nullable = true)
 ├── country: string (nullable = true)
 ├── avg_price_per_room: double (nullable = true)
 ├── email: string (nullable = true)
```

We can observe that the common features in both the schemas are the following:

['booking_status', 'lead_time', 'stays_in_weekend_nights', 'arrival_month', 'avg_price_per_room', 'market_segment_type', 'stays_in_week_nights', 'arrival_year']

2. **Missing Values and Null Value counts:**

There were 0% missing values or NULL values in the “customer_reservation” dataset. However, in the “hotel_booking” dataset, the feature “ country” has 405 or 0.51% of missing values and

NULL counts. Since it was much less in number and the feature was not in the common feature list, we kept it as is and moved forward with the analysis.

3. **Distinct values in each column:**

For both the datasets we counted the number of distinct values in each of the feature-columns. Some interesting results are as follows:

Table 2:

	Distinct Values	
Some Common Features	**customer_reservation**	**hotel_booking**
"booking_status"	{Not_Canceled, Canceled}	{1, 0}
"stays_in_weekend_nights"	{1,6,3,5,4,7,2,0}	{12,1,13,6,3,5,19,9,4,8,7,10,14,2,0,18, 16}
"arrival_month"	{ 1, 2, ….., 12}	{January, February, March, April, May, June, July, August, September, October, November, December}

While calculating the distinct values it showed us that the common column values needed to be standardized (later on) before merging.
The full list of distinct values per column for each dataset is shown in the attached .ipynb output.

4. **Summary Statistics for the Numerical features:**

Dataset: customer_reservation

Table 3:

Feature	**Mean**	**SD**	**Min**	**Max**
stays_in_weekend_nights	0.81	0.87	0	7
stays_in_week_nights	2.2	1.4	0	17
lead_time	85.23	85.93	0	443
arrival_year	**–**	**–**	2017	2018
arrival_month	7.42	3	1	12

arrival_date	15	8	1	31
avg_price_per_room	103.42	35	0	540

Dataset : hotel_booking

Table 4:

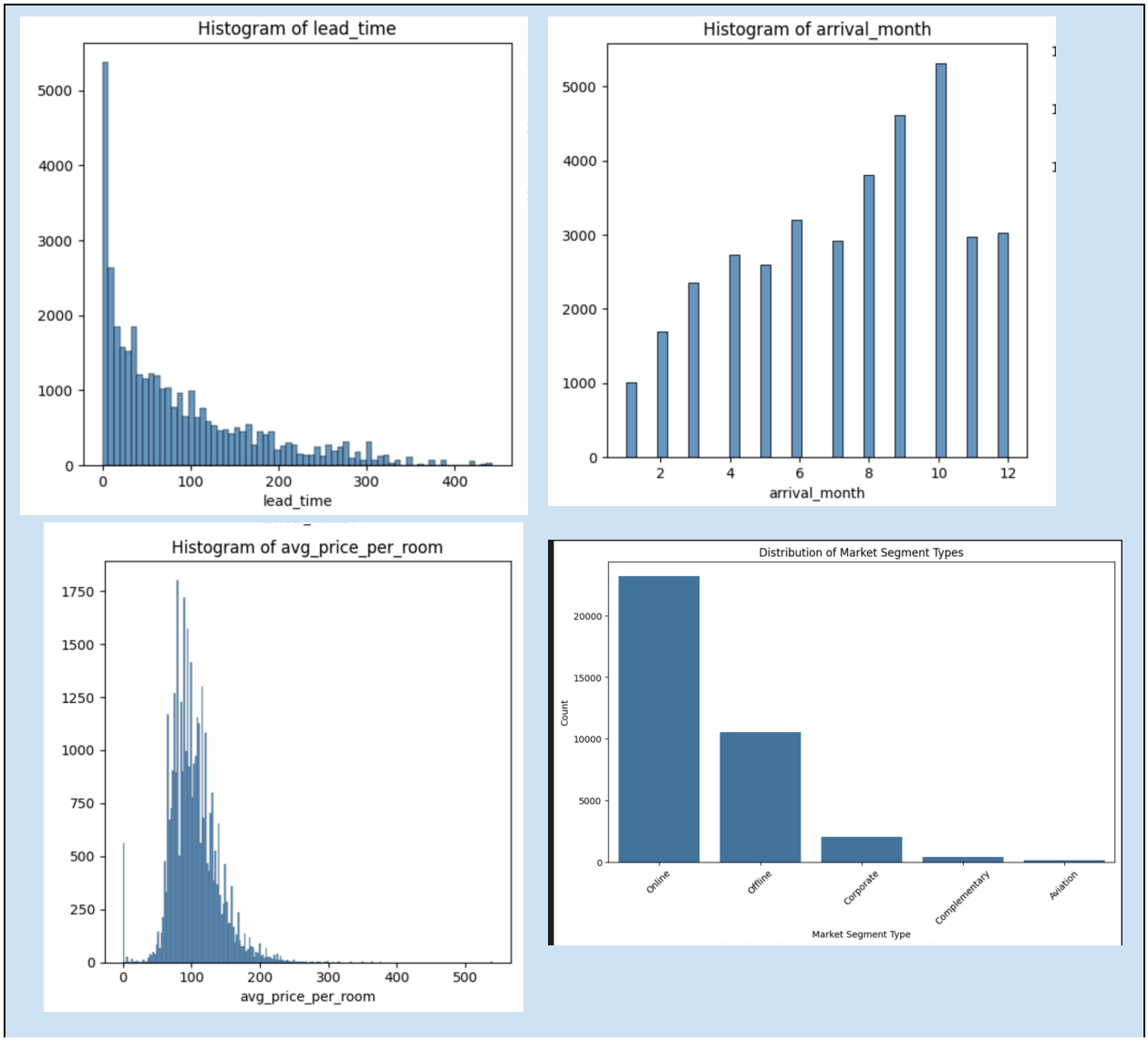
Feature	**Mean**	**SD**	**Min**	**Max**
stays_in_weekend_nights	0.90	0.98	0	19
stays_in_week_nights	2.44	1.87	0	50
lead_time	101.1	106.2	0	737
arrival_year	–	**–**	2015	2016
arrival_date_day_of_month	15.83	8.77	1	31
arrival_date_week_number	31.577	13.33	1	53
avg_price_per_room	95.21	48.30	0	5400

Some Relevant Observations from the Tables:
The above tables show that the data in the customer_reservation database is from 2017 and 2018, whereas the data in the hotel_booking database is from 2015 and 2016. So as expected the average room price increased from $95 to $103, however the standard deviation has decreased, which translates to the fact the distribution of the data has become a bit more "concentrated about the mean". There is a possible outlier in the avg_price_per_room in the hotel_booking database ($5400).

5. **Data Visualization:**
Here we have shown the histogram plots for some of the numerical feature columns and barplot for a categorical feature column. The full list of the figures are attached in the .ipynb notebook output.

Database : "customer_reservation"

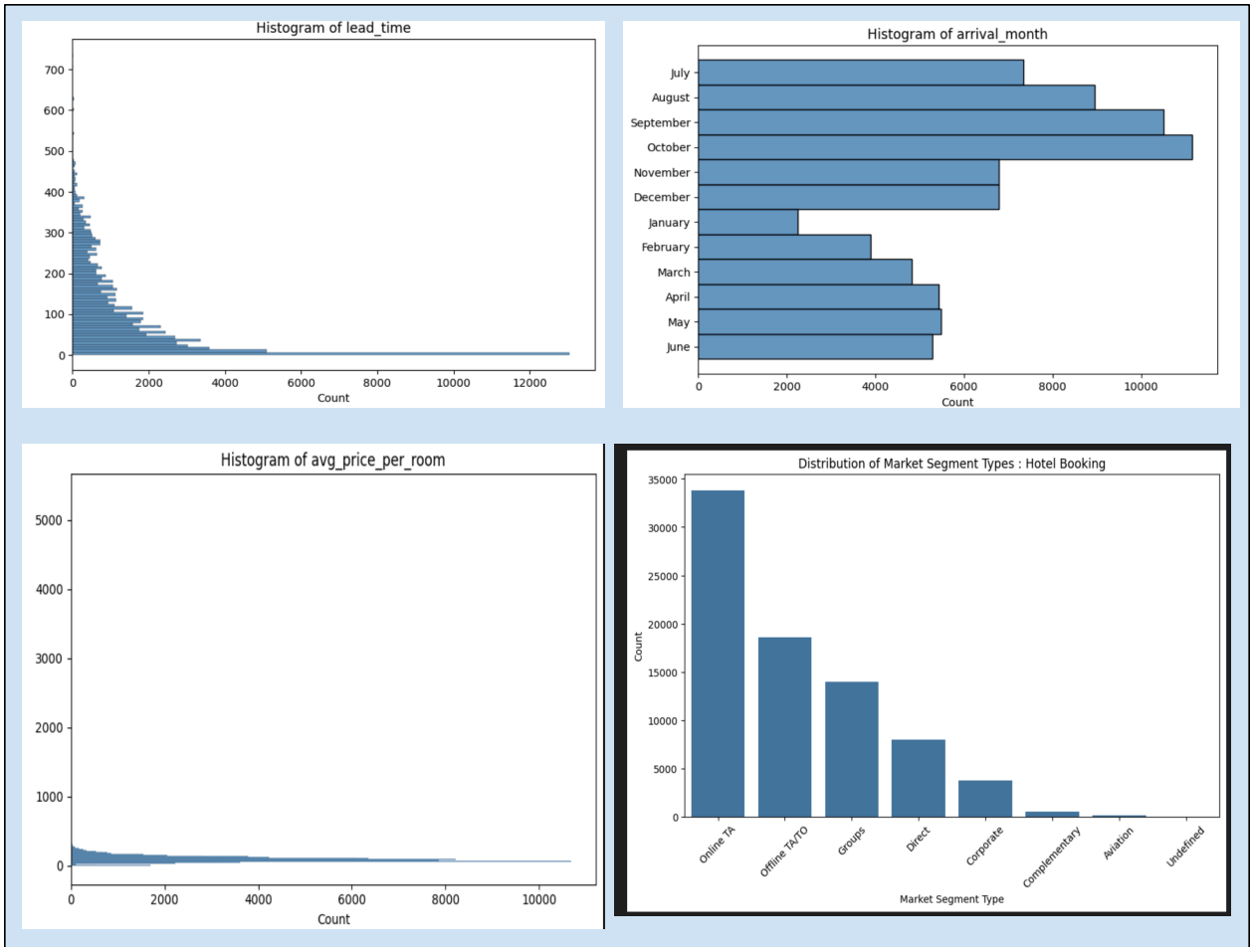
Observations:
We can observe that the "lead_time" is heavily right skewed, which means that most of the people did not wait long between the two bookings they made. Next, we can observe that the month of October and the Online market segment had the highest bookings. The average room price is somewhat normally distributed with some outlier values in the lower ranges.



Database: “hotel_booking”

Observations:
Here also the lead time is heavily right skewed which is consistent with the behaviour discussed above. The average room price as expected has an outlier. The media price is $89. The highest booking happened in the month of October and in the OnlineTA market segment type, which is also consistent with the results found in the database “customer_reservation”.

The distribution of arrival_date is almost uniform in both the databases. The distribution of weekend_nights and week_nights, booking in both the databases show that, the week_nights are more normally distributed whereas the weekend_nights are right skewed, which makes sense because weekend consists of 0-2 days and week consists of 0-5 days.



Merging the Databases:

We merged the two databases based on the common columns mentioned previously. We standardized the columns

- “booking_status” in “customer_reservation”
- “arrival_month” and “market_segment_type” in “hotel_booking”

The merged dataset has the following schema: (Total rows = 114978)

```
root
 ├── booking_status: integer (nullable = true)
 ├── lead_time: integer (nullable = true)
 ├── stays_in_weekend_nights: integer (nullable = true)
 ├── arrival_month: string (nullable = true)
 ├── avg_price_per_room: double (nullable = true)
 ├── market_segment_type: string (nullable = true)
 ├── stays_in_week_nights: integer (nullable = true)
 ├── arrival_year: integer (nullable = true)
```

We saved three new data sets as the .csv file as well as .xslx files.

Project Contribution:

Bufan: did the initial coding in .py file, did and wrote the “Installation of Pyspark : Bufan” in the report, generated new datasets in .xlsx format.

Sayantika: made the .ipynb notebook, wrote the report, generated new datasets in .csv format.