1. **Introduction**

   Data has a big impact on the hotel booking sector. Nowadays there are so many websites that can help a person to obtain updated information about different hotels around the world and book accommodation for their trip. People want to know the best time to book their accommodation with all their preferences and lowest price.

   However, since this data are dispersed in so many places, it will take time to collect all the information from different places, analyse them and reach to a decision. This scientific report contains a dataset collected from three different websites: Booking.com, Expedia.com, and Hotels.com. This report provides a solution for distributed information problem by gathering the data in one place and also give necessary data visualizations that will even enhance the decision process.

2. **Data collection**

   The data set contains hotel booking information from booking.com, expedia.com and hotels.com websites. The city selected as a filtering feature is New york. The date chosen for the trip is starting from 1-10-2022 to 27-10-2022. Number of person is 1.

   The websites are selected based on the time they need to be scraped. Booking.com has a consistent selection attribute called datatest-id for each element. This makes the scraping process fast because almost all elements can be found using this attribute. In addition, it uses pagination to list the hotels in different page. For pagination, it has different offsets for each page number. Since the offset increase by consistent interval which is 25 it is easier to iterate the pages. Furthermore, booking.com provides a vast amount of results for our search criteria.

   The other two websites expedia.com and hotels.com, since they are part of same travel group companies, their website structure are the same. Because of this reason, only one method is required to scrape data from these websites with their URL as a parameter.

   Therefore, the selection criteria for the websites are time to scrape them and the number of libraries required.

   The filtering values which are the city and the trip date are chosen to achieve the number of accommodations at least 50.

   Although, the websites contain different information, the dataset contains only the name, the number of stars, the price, the distance from city center, the rating section including both numeric and label, the description, the URL for more information and image links.

```
[221] hotelbooking_df.columns[1:]

Index(['Name', 'Numeric Rating', 'Label Rating', 'Review Number',
       'Number of Stars', 'Original Prices', 'Discount Prices', 'Locations',
       'Distance From City Center', 'Description', 'Hotel Page Url',
       'Image Urls'],
      dtype='object')
```

Figure 1: Dataset columns

```
hotelbooking_df.head()
```

	Unnamed: 0	Name	Numeric Rating	Label Rating	Review Number	Number of Stars	Original Prices	Discount Prices	Locations	Distance From City Center	Description	Hot
0	0	Hyatt Place New York City/Times Square	3.80	Good	12580.0	4.0	210125.0	NaN	Hell's Kitchen, New York	17.0	You're eligible for a Genius discount at Hyat...	https://www.booking.com
1	1	Freehand New York	4.00	Very good	1169.0	4.0	312239.0	NaN	Gramercy, New York	31.0	Located in the former George Washington Hotel,...	https://www.booking.com/hot
2	2	Hyatt Place NYC Chelsea	3.95	Good	3304.0	4.0	265622.0	NaN	Chelsea, New York	28.0	You're eligible for a Genius discount at Hyat...	https://www.booking.com
3	3	The Manhattan at Times Square	2.95	Review score	3464.0	4.0	209982.0	279976.0	Manhattan, New York	7.0	You're eligible for a Genius discount at The ...	https://www.booking.co
4	4	Fairfield Inn by Marriott New York Manhattan/F...	3.85	Good	2209.0	3.0	233831.0	NaN	Wall Street - Financial District, New York	72.0	Fairfield Inn by Marriott New York Manhattan/F...	https://www.booking.com/h

Figure 2: The first five records of the dataset

3. Data cleaning and feature engineering

Eventhough, we collect the data ourselves, since the sources are different a few data cleaning and feature engineering processes are applied to the original data set. First one dataset is created using padas dataframe by combining those three separate datasets.

```
235] hotelbooking_df.shape

(224, 14)

hotelbooking_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 224 entries, 0 to 19
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Unnamed: 0                 224 non-null    int64
 1   Name                       224 non-null    object
 2   Numeric Rating             224 non-null    float64
 3   Label Rating               224 non-null    object
 4   Review Number              224 non-null    float64
 5   Number of Stars            224 non-null    float64
 6   Original Prices            224 non-null    float64
 7   Discount Prices            50 non-null     float64
 8   Locations                  224 non-null    object
 9   Distance From City Center  224 non-null    float64
 10  Description                184 non-null    object
 11  Hotel Page Url             224 non-null    object
 12  Image Urls                 224 non-null    object
 13  Price                      50 non-null     float64
dtypes: float64(7), int64(1), object(6)
memory usage: 34.4+ KB
```

Figure 3: Dataset information

Figure 3 depicts that there are **224** rows and **14** columns in the dataset. There are null values in three of the features, which are discount prices, prices and descriptions. The discount prices are not found in all of the hotels, thus except the 50 records of the data all of them have null values. 184 of the hotels have description, the others don't have.
Additionally, the numeric features: original prices, discount prices, distance from city center, and review number, they had alphanumeric values in the website. Since their numeric values are required for the EDA(Explatory Data Analysis), a function is applied to extract only the number value and convert it to float value.
For numeric rating value, there was inconsistency regarding the scale across the websites. Booking.com use 10 as rating scale while the others use 5, thus another function is applied to convert all of them to scale 5.
Regarding the price, as described before, since we have two columns for price which are original prices and discount prices, additional column was needed that would be derived from both columns. If a row contains a null value for the discount price then the original price will be used otherwise the price will be the discount price.

4. **Exploratory data analysis (EDA)**
   The first step taken for the EDA section was finding a correlation between the numeric features, which are `'Numeric Rating'`, `'Review Number'`, `'Number of Stars'`, `'Distance From City Center'`, `'Price'` and Heatmap is used to show their relation.

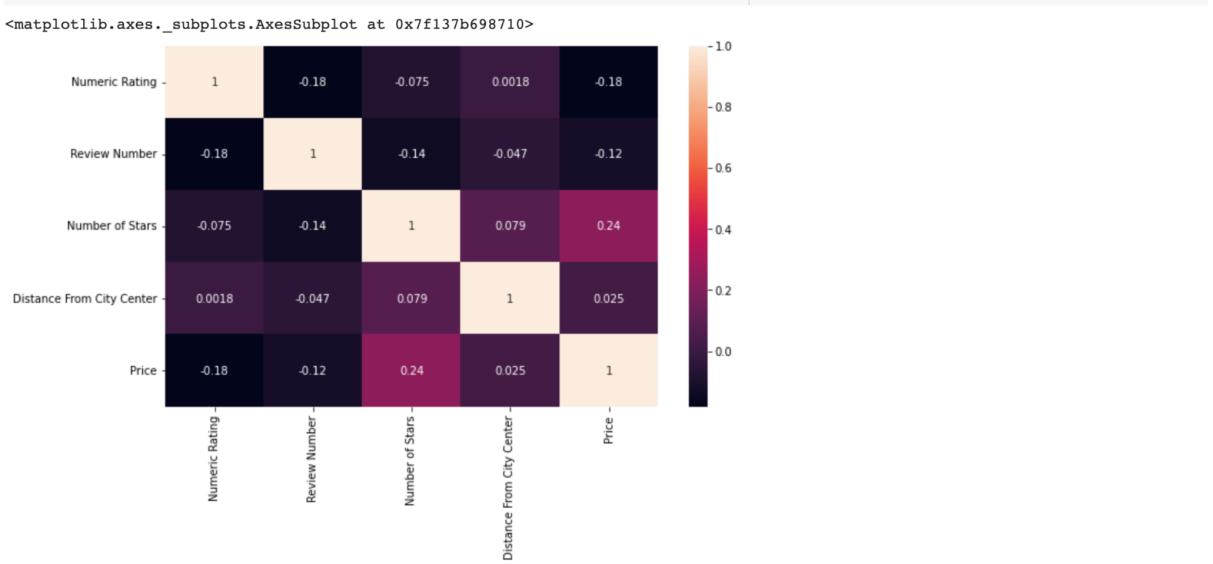

Figure 4: Heatmap figure, numeric features correlation

As the picture shows, most of the features don't have correlation. However, number of stars and price have a positive correlation. The price for more number of stars is expensive than the hotels with less number of stars.

Next we can see how many percent of the hotels rated as very good, good or the other label rating. We use horizontal bar chart.(figure 5)

```
# The percentage of each label rating category.
hotelbooking_df['Label Rating'].value_counts(normalize=True)

hotelbooking_df['Label Rating'].value_counts(normalize=True).plot.barh()
plt.show()
```

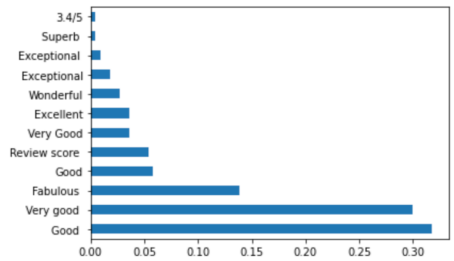

Figure 5: Label rating percentage

The other analysis point is related to how the label rating affects the price of the hotels. The data shows some of the hotels with very good label rating has expensive prices.

```
plt.rcParams["figure.figsize"] = (15,5.5)
hotelbooking_df.plot.scatter(x="Label Rating",y="Price")
plt.show()
```

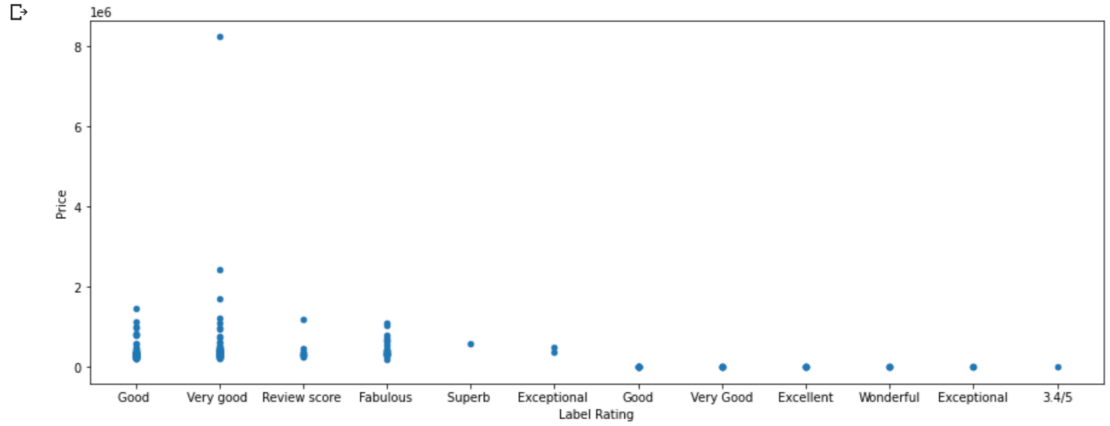

Figure 6: Label rating vs price scatter plot

Figure 6 also depicts there is an outlier with very good label status with the highest price.

5. Conclusion

The dataset is collected from three websites using python requests method to send request and Beautiful to access the page source elements. While scraping the data there were some troubles. The first one is selection of the website, since there are so many websites that provide these kind of data choosing the best website to get accurate data with the time limit was challenging. Finally, since the combination of these three websites with the given filtering criteria give us more than 50 accommodations, thus they have been selected. The other problem was the internationalization feature of the websites. These websites provide their data based on the location of the user, but with python request module they sometimes send different data(for-example: price in different currency, or distance in different unit). Initially, the problem was not

clear, however as long as the data is scraped it doesn't matter in which unit it is provided, thus leaving as it is the solution. The dynamic data loading in the website using javascript was the other problem in some of the websites. Since Beautifulsoup doesn't support this feature instead of going for the other libraries like selenium, changing the websites were the solution.

6. Code Usage Guide
    a. The code is written in google colab
    b. Running the third cell will import all the required libraries
    c. To store the csv file and the code, google drive is being used
        i. Running the fourth cell will mount drive
        ii. Mini-project-1 folder is created inside `My Drive` folder