Hotels in KSA

1. Introduction

Saudi Arabia has moved to this market itself as a global tourism destination with speed and ambition.

The KSA, the region's largest outbound tourism market, has sharpened its focus on further developing the country's tourism offerings of late. The tourism sector comes as a result of Saudi Vision 2030, a strategic masterplan encompassing a wide range of social and economic reforms across a multitude of sectors, aimed at reducing the country's dependence on oil and diversifying into more sustainable sources. A number of ambitious megaprojects have taken shape since that launch,

So Have you ever wondered what the highest city was booked? And Is the the highest customer rating are the ones with highest prices? What type of room was has the highest price ? This case study help you explore those questions !and more

Also If u think to investment in this market maybe I will show u some interesting information so be ready!

2. Data Review

But First lets me introduce our data to you , the dataset wes from *Kaggle* its web-scrapping dataset from Booking.com. it is about data of 1025 hotels in Saudi Arabian, it have 21 variables and 1025 observations

3. Data Preprocessing

As you can see, we have 10 features with missing values which is:

```
Type_of_room              1
Bed_type                 19
Customers_Rating         69
Customers_Review         69
Review_title             69
Canelation              179
Property_Demand         275
reservations_Payment    290
Credit_card             321
Breakfst_included       941
```

So, the total amount of missing values is 2233.
Here is the method we use for each column:

- 'Breakfst_included':

We decide to delete 'Breakfst_included' because it has more than 90% missing values

```
df = df.drop('Breakfst_included', axis = 1)
#because it has more than 90% missing values
```

- 'Type_of_room':

For the missing values in 'Type_of_room' column, replace it with mode (value that appears most often)

```
df['Type_of_room'].fillna(df.Type_of_room.mode().to_string(), inplace=True)
```

- 'Customers_Review' & 'Customers_Rating':

We use ‘mean’ to get the average

```
df['Customers_Review'].fillna(df.Customers_Review.mean(), inplace=True)
```

```
df['Customers_Rating'].fillna(df.Customers_Rating.mean(), inplace=True)
```

- The rest columns:

We use Fill values backward

```
df["Review_title"].fillna( method ='bfill', inplace = True)
df["Canelation"].fillna( method ='bfill', inplace = True)
df["Property_Demand"].fillna( method ='bfill', inplace = True)
df["reservations_Payment"].fillna( method ='bfill', inplace = True)
df["Credit_card"].fillna( method ='bfill', inplace = True)
df["Bed_type"].fillna( method ='bfill', inplace = True)
```

- 'Bed_type':

It had symbols and line. So we split it to beds numbers and bed details And then for the bed details and since they were not organised, we split them again and put them in for loop. And as the for loop iterates, it will check the number for each bed type (for each record in the dataframe) and then append it to a new dataframe which is then merged to the original dataframe.

```
## Splitting the bed
df2['NumberOfBeds'] , df2['BedsDetails']  = df2['Bed_type'].str.rsplit('\n', 1).str

## Splitting the City and then filling Nulll values in the City to be the values from the other colum same row.
#df2['BedTypeDetails'] , df2['BedType']  = df2['Bed_type'].str.rsplit('\n', 1).str
df2.NumberOfBeds =  df2.NumberOfBeds.apply(lambda x: x.replace('bed',''))
df2.NumberOfBeds =  df2.NumberOfBeds.apply(lambda x: x.replace('s',''))
#df2.NumberOfBeds.value_counts()


df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace('(',''))
df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace(')',''))

df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace('singles','single'))
df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace('beds','bed'))
df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace('doubles','double'))
df2.BedsDetails =  df2.BedsDetails.apply(lambda x: x.replace(u'\xa0', u'')) ### ['2 single', '\xa01 extra-large double']

#df2.BedsDetails.value_counts()
# extra-large double ,    sofa bed,    large double,     double,     single,  bunk bed --> 6 Types
columns2 = ["extra-large double" ,    "sofa bed",    "large double",     "double",     "single",  "bunk bed" ]

#df4 = df2['BedsDetails'].str.split(',', expand=True)
#df4 = pd.get_dummies(df2.BedsDetails).reindex(columns = columns2, fill_value=0)

# df4 = df2.join(pd.DataFrame(df2.pop('BedsDetails').values.tolist()))
```

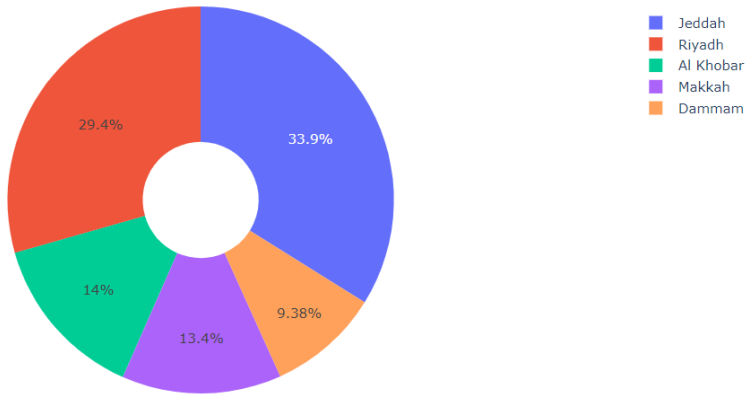
```
#for row in df2.itertuples(index=True, name='Pandas'):
#    arrays = (row.BedsDetails).str.split(',')
#    print(arrays)

df_bedType = pd.DataFrame(columns=[ "ID",  "extra-large double" ,    "sofa bed",    "large double",     "double",     "single",  "bunk

for index, row in df2.iterrows():
    rowArray= (row["BedsDetails"]).split(',')
    #print(index)
    sofa= 0
    XlargeDouble= 0
    largeDouble= 0
    double= 0
    single= 0
    bunkBed = 0

    #print(arrays)
    for i in rowArray:
        bedTypeNumbers, bedType = i.split(' ', 1)
        #print(bedTypeNumbers)
        if bedType == 'single':
            single = bedTypeNumbers
        elif bedType == 'extra-large double':
            XlargeDouble = bedTypeNumbers
        elif bedType == 'sofa bed':
            sofa = bedTypeNumbers
        elif bedType == 'large double':
            largeDouble = bedTypeNumbers
        elif bedType == 'double':
            double = bedTypeNumbers
        else:
```

```
            bunkBed = bedTypeNumbers
        new_row = pd.Series(data={'ID': index   , "extra-large double":  XlargeDouble ,    "sofa bed":  sofa    ,    "large double": lar
                                  "double":  double      ,     "single":   single    ,  "bunk bed":   bunkBed        })

        #append row to the dataframe
        df_bedType = df_bedType.append(new_row, ignore_index=True)

df_bedType
```

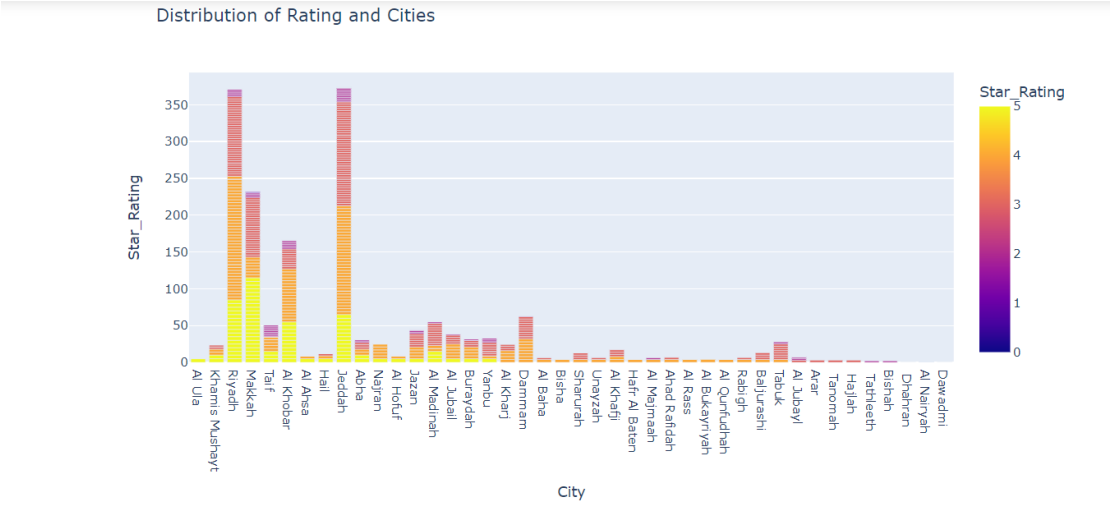
	ID	extra-large double	sofa bed	large double	double	single	bunk bed
0	0	0	0	3	0	0	0
1	1	1	0	0	0	0	0
2	2	0	0	0	1	0	0
3	3	0	0	0	0	2	0
4	4	0	0	0	1	0	0
...	...	...	...	...	...	...	...
1147	1020	0	0	0	0	2	0
1148	1021	0	0	0	1	0	0
1149	1022	0	0	0	1	0	0
1150	1023	1	0	0	0	0	0
1151	1024	1	0	0	0	0	0

1152 rows × 7 columns

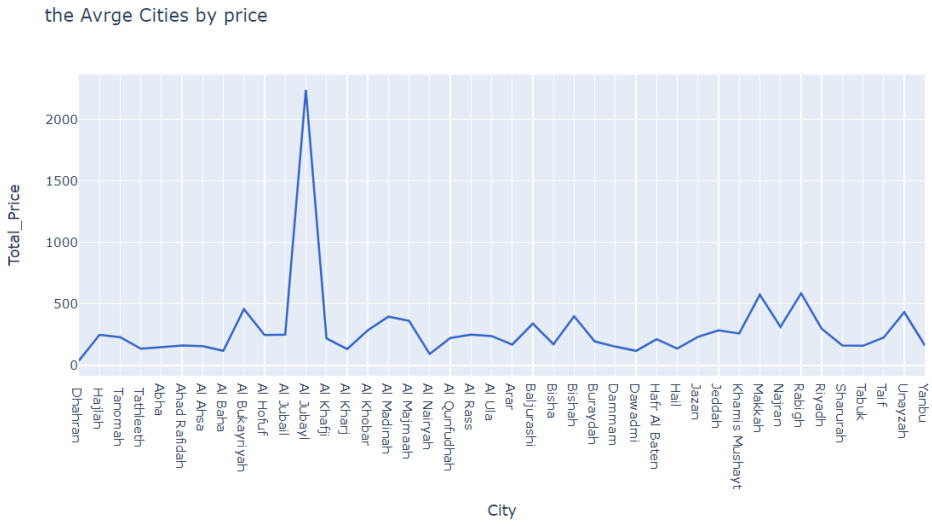
4. Data Exploration



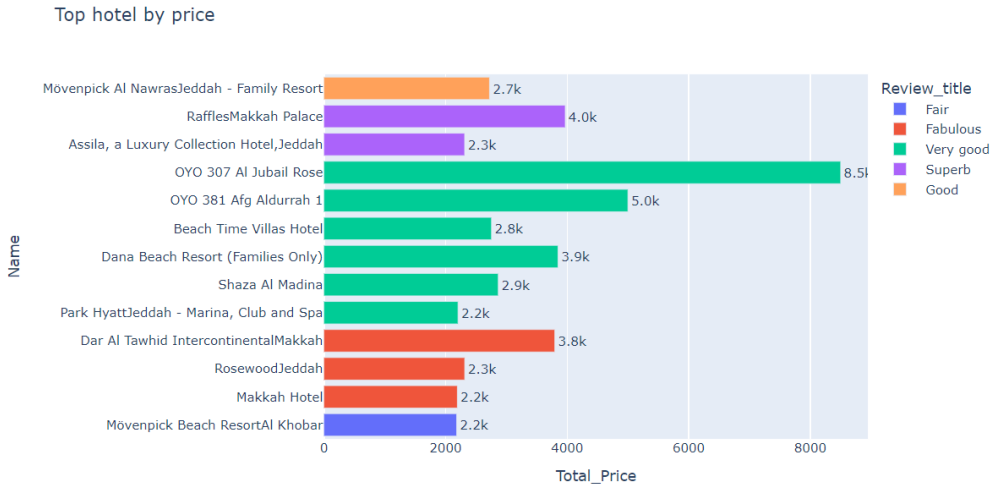
This is the Top 5 cities by number of hotels So Jeddah come first then Riyadh then Al Khobar then Makkah then al Damaam



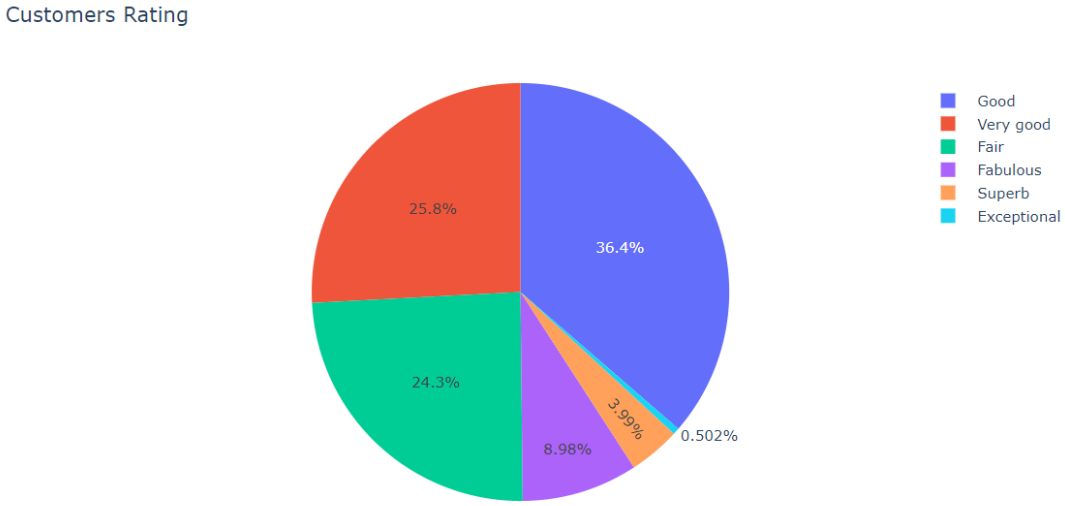
So here the highest city by rating is Jeedah then AL Riyadh then Makkah but the highest rating doesn't mean it's good because it has a lot of 3 rating



Al Jubail City have the highest average of price because it has one of the most price Expensive hotel



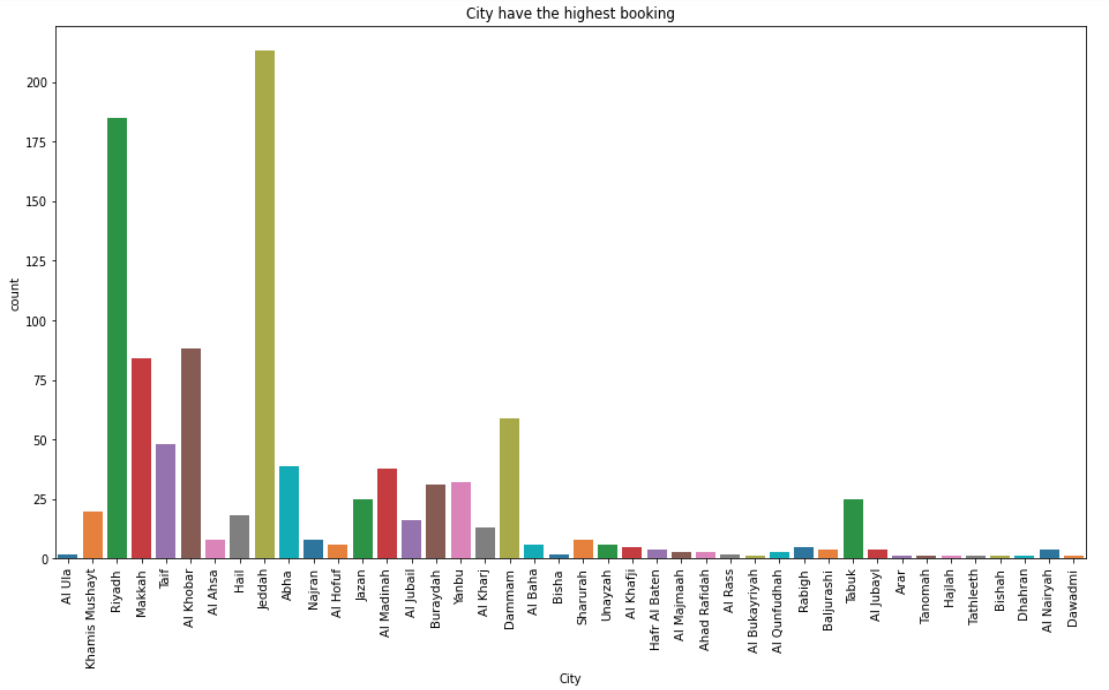
And if u want a luxury experience this is the Top hotel by price or if u think to investment in this market



Here is the general look of customers rating the most of customers are satisfied with the services

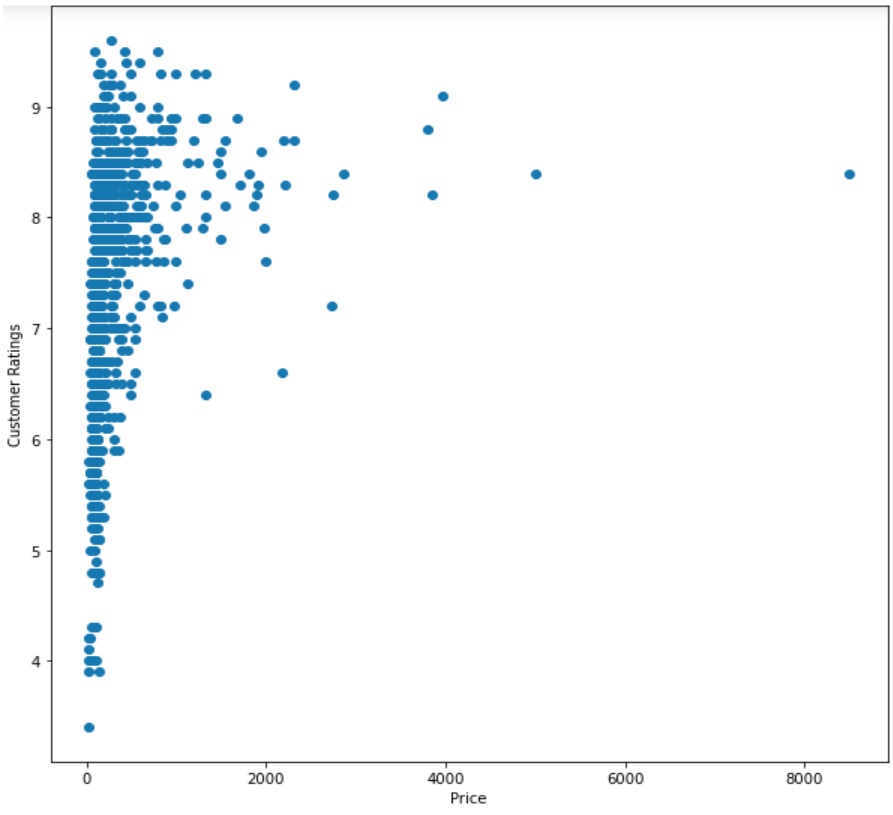
Extract the information from our data and try to answer our questions.

1-What is the highest hotels booking between Cities?



More than 200 of the booked hotels was in Jeddah.

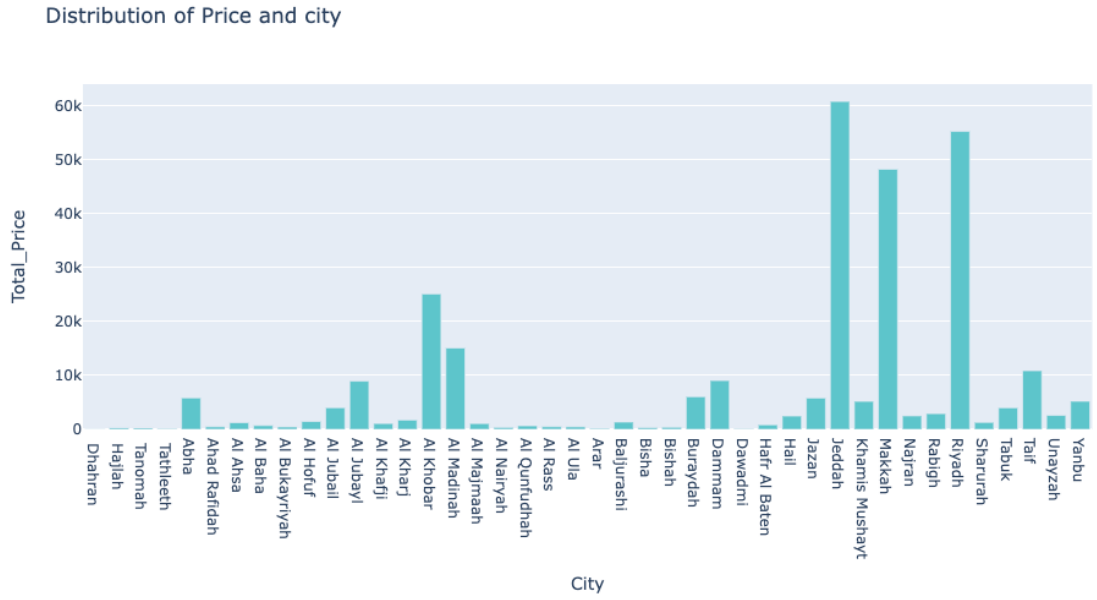
2- What is the highest customer rating in terms of price?



We cannot say that the highest customer rating hotels are the ones with highest prices.

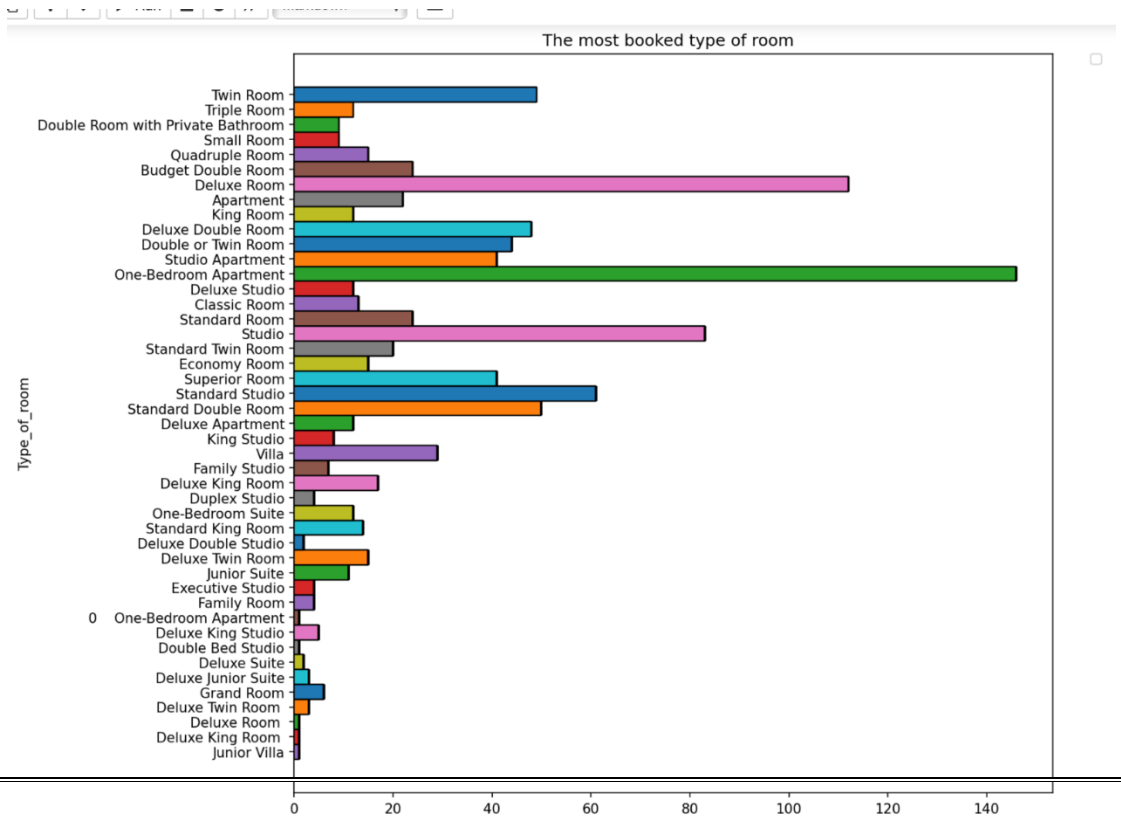
but we can say that the generally, the high-priced hotels have 6+ customers' ratings.

3-Which city is the highest price by total price?

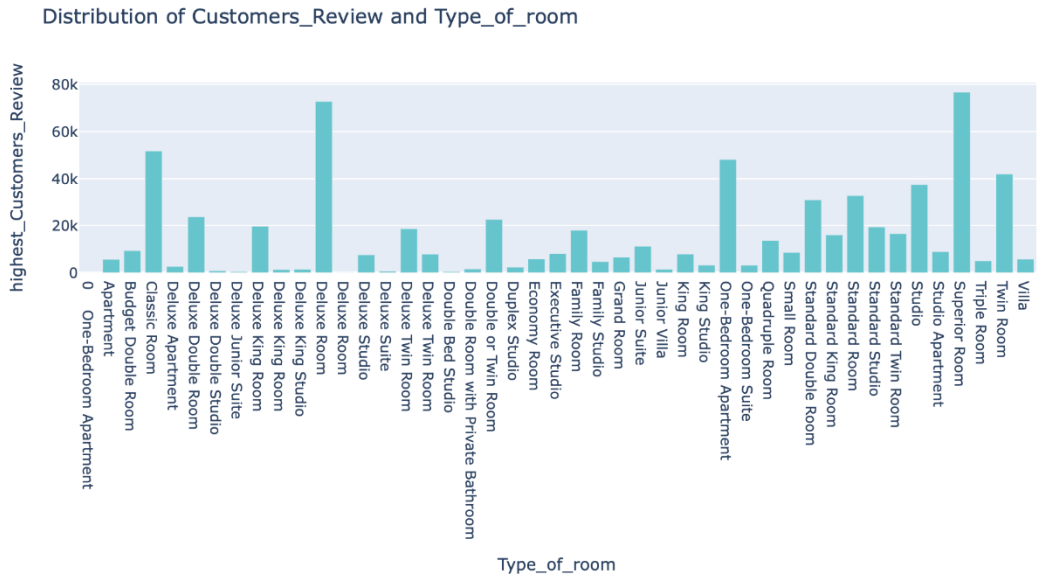


We note in this graph that the highest total price is in the following main cities: The capital Riyadh, Jeddah and the holy capital Makkah.

4-What is The most booked type of Room ?



5-What is The Highest customers review by the Type of Room ?



The Superior Room is the highest customers review and also the type of Deluxe Room is so near from the highest one .

The Superior Room has 80k reviews and the Deluxe Room so near from 80k reviews . so these two types is the highest reviews .

5. Building Classification Models

We started off with encoding our features:

1. One hot encoding: City and Type_of_room.
2. Label Encoding: Review Title

After that, we have split our dataset to X and y (target), dropped unneeded columns and then split them again to X train and test and y train and test while making sure that our test dataset size is 20% and 80% is for the training set. Next, we used standard scaler to scale our features.

Out of the many regressors we have tried, two main regressor models are noted and compared and they are the XGBoost regressor and Random Forest Regressor.

XGBoost regressor gave us the highest value for r^2 (Coefficient of determination) which is 74% and as we know the higher value of R2 is the better. RMSE is 211% and MAE 100%, the lowest values of all the models we have tried.

We have also tried Grid Search to tune our XGBoost parameters, however that did not give us higher results than our original result. Not only that, but we have also tried PCA to lower the dimensionality of our data from 102 to 48 and that also did not work, R2 score was very low.

Index	ActualValue	PredictionXGboost	PredictionRandomFores	PercentageChangeXG	PercentageChangeRF
946	115	110.28448	106.7	4.275781039	7.778819119
363	143	151.9129	139.3	-5.867113764	2.656137832
1095	129	176.67471	141.3	-26.98445764	-8.704883227
678	53	81.99235	65.9	-35.3598214	-19.57511381
311	83	193.15742	161	-57.02986824	-48.44720497
783	93	106.90971	105.8	-13.01071087	-12.09829868
733	75	98.778465	107.6	-24.07251946	-30.29739777
224	419	329.6227	232.6	27.11502751	80.13757524
491	299	293.44827	252.4	1.891892988	18.46275753
202	375	375.22513	365.9	-0.05999816	2.487018311

The top 10 rows from the ML Models Predictions.

6. Results

- most of customers are satisfied with the services.
- Al Jubail City have the highest average of price.
- the highest rating doesn't mean the hotels services its good.
- The highest total price city Jeedah.
- The Superior room was the Highest customer's review.
- One bedroom Apartment was the Highest price room.