**PROPERTY LISTING ANALYSIS**

Deepanshu Singh (pd13\_057)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You are an Analyst working for a Property Rental company. The company wants to explore the trends for different property types (Room Types) and their listed price across a variety of metrics. Thus, you are asked to understand the data and come up with the below analysis:

1. Analyse **different metrices** to draw the distinction between the **different types of property** along with their price listings (bucketize them within 3-4 categories basis your understanding):

To achieve this, you can use the following metrics and explore a few yourself as well.

Availability within 15,30,45, etc. days, Acceptance Rate, Average no of bookings, reviews, etc.

1. PRICE METRICS ON PROPERTY TYPES:

**QUERY:**

Create view PT\_Budget as

Select \*, Case when Avg\_Price< =100 Then 'Affordable'

when Avg\_Price > 1000 Then 'Luxurious'

when Avg\_Price between 500 and 1000 Then 'Ultra Premium'

else 'Premium'

end as 'Stay'

FROM

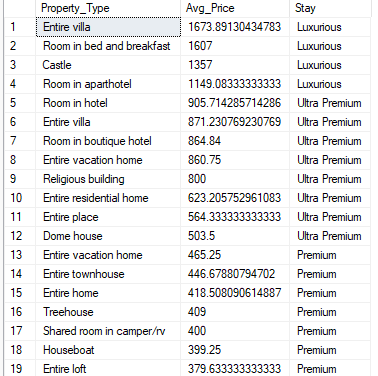
(Select Property\_Type, avg(price) as Avg\_Price from listing\_dallas\_df group by property\_type

UNION

Select Property\_Type, avg(price) as Avg\_Price from listing\_austin\_df group by property\_type)Ap;

---------------------------------------------------------------------------Select \* from PT\_Budget;

**OUTPUT(Sample):**



RATIONALE behind query:

* To get average price of all property types.
* A sub-query to UNION desired output from 2 cities.
* CASE WHEN for bucketizing budget type stays based on avg price.
* Created a VIEW to use query later.

1. ACCEPTANCE METRICS ON PROPERTY TYPES

QUERY:

Create VIEW Avg\_ACR as

Select Property\_Type, Avg(Avg\_AR) as Avg\_AcpR from

(Select listD.Property\_Type, AVG(hostD.host\_acceptance\_rate) as Avg\_AR from listing\_dallas\_df as listD join host\_dallas\_df as hostD

ON listD.host\_id = hostD.host\_id

GROUP BY listD.Property\_Type

UNION

Select listA.Property\_Type, AVG(hostA.host\_acceptance\_rate) as Avg\_AR from listing\_austin\_df as listA

join host\_austin\_df as hostA

ON listA.host\_id = hostA.host\_id

GROUP BY listA.Property\_Type)Avg\_AR

GROUP BY Property\_Type

---------------------------------------------------------------------------

Select \* from Avg\_ACR;

------------------------------ FINAL QUERY --------------------------------

Select Property\_Type,Avg\_AcpR,

Case when Avg\_AcpR between 90 and 100 then 'Great'

when Avg\_AcpR between 70 and 89 then 'Good'

when Avg\_AcpR between 40 and 89 then 'Average'

when Avg\_AcpR < 40 then 'Poor'

Else 'N.A.'

End as Acceptance

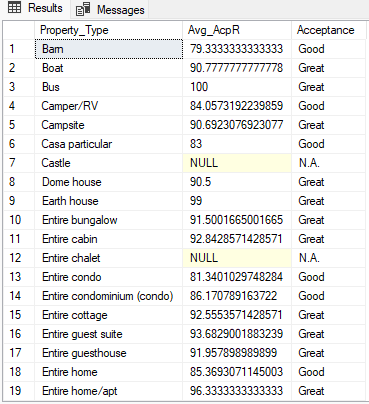
From

(Select B.\*,A.Avg\_AcpR from PT\_Budget as B

join Avg\_ACR as A

ON B.property\_type=A.property\_type)AR;

OUTPUT:



RATIONALE behind query:

* To get average
* In Sub-query, JOINed listings and hosts, followed by UNION for data of both cities, with OUTER query grouping avg of property types to eliminate duplicate property types.- Created a VIEW to use this query later
* JOIN 2 Views based on Property type in inner query and bucketized acceptance rates with a comment by using CASE WHEN.

1. RATINGS METRICS ON PROPERTY TYPES

QUERY:

Select \*, Case when Avg\_of\_rating = 5 then 'Extraodinary'

when Avg\_of\_rating > 4.5 then 'Excellent'

when Avg\_of\_rating > 4 then 'Good'

when Avg\_of\_rating > 3 then 'Average'

when Avg\_of\_rating <= 3 then 'Poor'

else 'N.A.'

eND as 'Rating'

from

(Select Property\_Type, Avg(Avg\_) as Avg\_of\_rating from

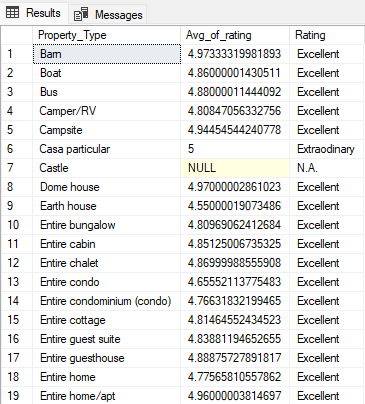
(Select Property\_Type, Avg(review\_scores\_rating) as Avg\_ from listing\_austin\_df group by Property\_Type

UNION

Select Property\_Type, Avg(review\_scores\_rating) as Avg\_ from listing\_dallas\_df group by Property\_Type)Av\_R

Group by Property\_Type)Avg\_Rate;

OUTPUT:



RATIONALE behind query:

* To bucketize rating categories, used a CASE WHEN.
* In the inner most query, extracted grouped avg ratings according to property types, UNIONed both Cities.
* Outer groups property types again to eliminate repetitive values.
* Outer most query is all queries and CASE WHEN bucketization.

1. BOOKING METRICS ON PROPERTY TYPES

QUERY:

Select \*, Case when Number\_of\_bookings> 100000 then 'Most Bookings'

when Number\_of\_bookings < 10000 then 'Least Bookings'

else 'Moderate Bookings'

end as Bookings

from

(Select Property\_Type, Sum(No\_of\_bookings) as Number\_of\_bookings from

(Select listA.property\_type,count(available) as No\_of\_bookings from listing\_austin\_df as listA

join df\_austin\_availability as AvailA

on listA.id=AvailA.listing\_id

where AvailA.available=0

Group by listA.property\_type

UNION

Select listD.property\_type,count(available) as No\_of\_bookings from listing\_dallas\_df as listD

join df\_dallas\_availability as AvailD

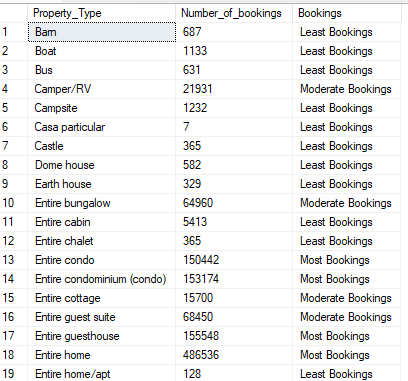
on listD.id=AvailD.listing\_id

where AvailD.available=0

Group by listD.property\_type)B

GROUP BY Property\_Type)Bookings;

OUTPUT(Sample):



RATIONALE:

* To analyse booking numbers across property types (Assumed Availability=0 as Booking)
* In inner query, created a JOIN between listings and availability, grouped count of bookings by property types. UNIONed both cities.
* In OUTER query, Bucketized booking number to perform analysis.

1. Study the **trends of the different categories and provide insights** on same
2. Price trends across broad listing categories:

QUERY:

Select Room\_Type, Min(Min\_P) as Min\_Price, Max(Max\_P) as Max\_Price, Avg(Avg\_P) as Avg\_Price from

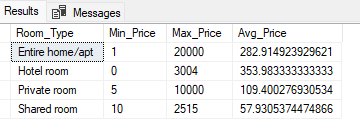
(Select Room\_Type, min(price) as Min\_P, max(price) as Max\_P, avg(price) as Avg\_P from listing\_dallas\_df group by room\_type

UNION

Select Room\_Type, min(price) as Min\_P, max(price) as Max\_P, avg(price) as Avg\_P from listing\_austin\_df group by room\_type)PT

Group by Room\_Type;

OUTPUT:



RATIONALE:

* To group price trends by category and plot those trends on charts.
* UNIONed records from both cities.

1. Acceptance trends across broad listing categories

QUERY:

Select Room\_Type, Avg(Avg\_) as Avg\_AR from

(Select listD.Room\_Type, AVG(hostD.host\_acceptance\_rate) as Avg\_ from listing\_dallas\_df as listD

join host\_dallas\_df as hostD

ON listD.host\_id = hostD.host\_id

GROUP BY listD.Room\_Type

UNION

Select listA.Room\_Type, AVG(hostA.host\_acceptance\_rate) as Avg\_ from listing\_austin\_df as listA

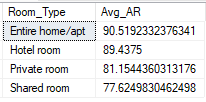
join host\_austin\_df as hostA

ON listA.host\_id = hostA.host\_id

GROUP BY listA.Room\_Type)Acr\_RT

Group by Room\_Type;

OUTPUT:



RATIONALE:

* To group acceptance across categories.
* JOINED listings and hosts with grouping by acceptance rate for room\_type(categories).
* Grouped in OUTER query to eliminate repetitive categories.

1. Using the above analysis, identify top 2 crucial metrics which makes different property types along their listing price stand ahead of other categories
2. A correlation between 2 important metrics price and ratings:

QUERY:

Create View PT\_Ratings as

Select \*, Case when Avg\_of\_rating > 4.5 then 'Excellent'

when Avg\_of\_rating > 4 then 'Good'

when Avg\_of\_rating > 3 then 'Average'

else 'Poor'

eND as 'Rating'

from

(Select Property\_Type, Avg(Avg\_) as Avg\_of\_rating from

(Select Property\_Type, Avg(review\_scores\_accuracy) as Avg\_ from listing\_austin\_df group by Property\_Type

UNION

Select Property\_Type, Avg(review\_scores\_accuracy) as Avg\_ from listing\_dallas\_df group by Property\_Type)Av\_R

Group by Property\_Type)Avg\_Rate;

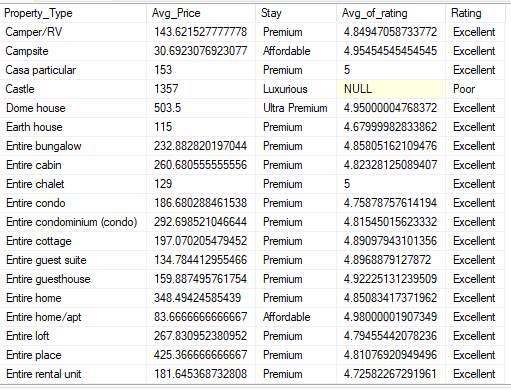
----------------------------- FINAL QUERY ---------------------------------

Select B.\*,R.Avg\_of\_rating,R.Rating from PT\_Budget as B

join PT\_Ratings as R

on b.property\_type=r.property\_type;

OUTPUT:



RATIONALE:

* To perform an analysis on how pricing and Stay type make an impact on rating
* Created a VIEW for Rating into and JOINED it with earlier view

1. Analyse how does the comments of reviewers vary for listings of distinct categories (Extract words from the comments provided by the reviewers)

**Variation of positive and negative comments(based on given key words) across categories:**

QUERY:

Create view list\_revA as

Select listA.room\_type, revA.comments from listing\_austin\_df as listA

join review\_austin\_df as revA

on listA.id=revA.listing\_id;

Select \* from list\_revA;

-----------------------------------------------------------------------

Create view list\_revD as

Select listD.room\_type, revD.comments from listing\_dallas\_df as listD

join review\_dallas\_df as revD

on listD.id=revD.listing\_id;

Select \* from list\_revD;

----------------------------------------------------------------------

Create View a\_comments as

WITH P\_A as

(Select Room\_Type, count(comments) as Postive\_comments from list\_revA where

comments like '%excellent%' or comments like '%great%' or

comments like '%Awesome%' or comments like '%Good%' and comments like '%Comfort%'

group by Room\_Type),

N\_A as

(Select Room\_Type, count(comments) as Negative\_comments from list\_revA where

comments like '%Bad%' or comments like '%Worst%'

or comments like '%Not good%' or comments like '%Unsatisfac%'

group by Room\_Type)

Select P\_A.\*,N\_A.Negative\_Comments from P\_A

full outer join N\_A

ON P\_A.Room\_Type=N\_A.Room\_Type;

Select \* from a\_comments;

-------------------------------------------------------------------------

Create View d\_comments as

WITH P\_D as

(Select Room\_Type, count(comments) as Postive\_comments from list\_revD where

comments like '%excellent%' or comments like '%great%' or

comments like '%Awesome%' or comments like '%Good%' and comments like '%Comfort%'

group by Room\_Type),

N\_D as

(Select Room\_Type, count(comments) as Negative\_comments from list\_revD where

comments like '%Bad%' or comments like '%Worst%'

or comments like '%Not good%' or comments like '%Unsatisfac%'

group by Room\_Type)

Select P\_D.\*,N\_D.Negative\_Comments from P\_D

full outer join N\_D

ON P\_D.Room\_Type=N\_D.Room\_Type;

--------------------------------FINAL QUERY--------------------------------

Select Room\_Type, Sum(Postive\_comments) as Comments\_Positive, Sum(Negative\_Comments) as Comments\_Negative

from

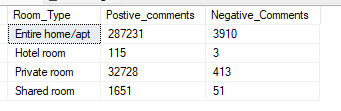
(Select \* from d\_comments

UNION

Select \* from a\_comments)C

Group by Room\_Type;

OUTPUT:



RATIONALE:

* To get the count of comments into positive and negative across categories based on set of key words.

1. Analyze if there is any correlation between property type and their availability across the months

QUERY:

Create view list\_availA as

Select listA.\*, availA.available, availA.date from listing\_austin\_df as listA

join df\_austin\_availability as availA

on listA.id=availA.listing\_id;

--------------------------------------------------------------------------------

Create view list\_availD as

Select listD.\*, availD.available, availD.date from listing\_dallas\_df as listD

join df\_dallas\_availability as availD

on listD.id=availD.listing\_id;

---------------------------------------------------------------------------------

Create view PT\_Monthly\_Availability as

With m\_A as

(Select property\_type, concat\_ws('-',datename(month,date), year(date)) as Month\_Year, count(available) as available\_listings from list\_availA

where available=1

group by property\_type,concat\_ws('-',datename(month,date), year(date))),

m\_D as

(Select property\_type, concat\_ws('-',datename(month,date), year(date)) as Month\_Year, count(available) as available\_listings from list\_availD

where available=1

group by property\_type, concat\_ws('-',datename(month,date), year(date)))

Select \* from m\_A

union

Select \* from m\_D;

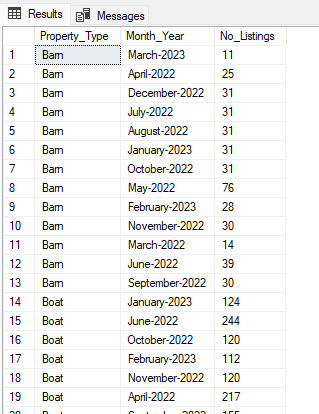
-----------------------------FINAL QUERY----------------------------------------

Select Property\_Type, Month\_Year, Sum(available\_listings) as No\_Listings from (Select \* from PT\_Monthly\_Availability)PMA

Group by Property\_Type, Month\_Year

Order by Property\_Type;

OUTPUT:



RATIONALE:

* To group Property\_type on monthly basis by number of available listings, so as to perform an analysis on output.

1. Analyze what are the peak and off-peak time for the different categories of property type and their listings. Do we see some commonalities in the trend or is it dependent on the category?

QUERY:

-------------------------------------------PEAK-----------------------------------

Select room\_type, Month , Year, Sum(Av) as Peak\_Counts from

(

Select room\_type, datepart(month,date) as Month , year(date) as Year, count(available) as Av from list\_availA

where available=0 group by room\_type, datepart(month,date),year(date)

UNION

Select room\_type, datepart(month,date) as month, year(date) as Year, count(available) Av from list\_availD

where available=0 group by room\_type, datepart(month,date),year(date)

)Avail

group by room\_type, Month , Year

order by room\_type;

-------------------------------------------OFF PEAK-------------------------------

Select room\_type, Month , Year, Sum(Av) as Off\_Peak\_Counts from

(

Select room\_type, datepart(month,date) as Month , year(date) as Year, count(available) as Av from list\_availA

where available=1 group by room\_type, datepart(month,date),year(date)

UNION

Select room\_type, datepart(month,date) as month, year(date) as Year, count(available) Av from list\_availD

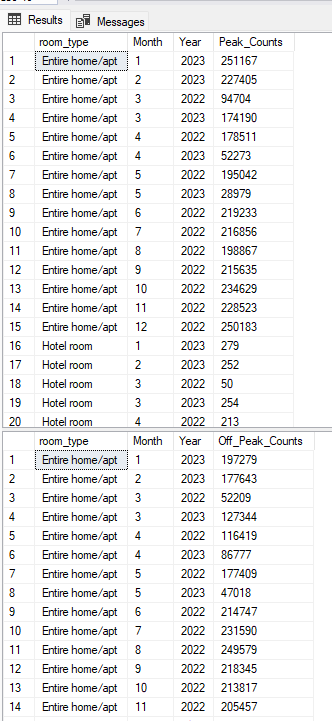
where available=1 group by room\_type, datepart(month,date),year(date)

)Avail

group by room\_type, Month , Year

order by room\_type;

OUTPUT:



RATIONALE:

* Assumed non availability as peak and availability as off peak.
* And grouped Property\_type on monthly basis by number of peak and off peak listings, so as to perform an analysis on output.

1. Using the above analysis, suggest what is the best performing category for the company

QUERY:

Select room\_type, Sum(Peak\_Counts) as No\_of\_Booking from

(Select room\_type, Month , Year, Sum(Av) as Peak\_Counts from

(

Select room\_type, datepart(month,date) as Month , year(date) as Year, count(available) as Av from list\_availA

where available=0 group by room\_type, datepart(month,date),year(date)

UNION

Select room\_type, datepart(month,date) as month, year(date) as Year, count(available) Av from list\_availD

where available=0 group by room\_type, datepart(month,date),year(date)

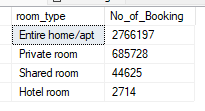
)Avail

group by room\_type, Month , Year)Cat\_Analysis

group by room\_type

order by Sum(Peak\_Counts) desc;

OUTPUT:



RATIONALE:

* To get number of bookings across category, so as to know best performing category for the company.
* Grouped Room\_Types by number of bookings;

1. Analyze the above trends for the two cities for which data has been provided and provide insights on comparison

Comparisional analysis on below metrics/trends between two cities across listing categories:

QUERY:

-i-Average Price

Select Room\_Type, avg(price) as Avg\_Price from listing\_dallas\_df group by room\_type;

Select Room\_Type, avg(price) as Avg\_Price from listing\_austin\_df group by room\_type;

-ii-Ratings

Select Room\_Type, Avg(review\_scores\_rating) as Average\_Ratings from listing\_austin\_df group by Room\_Type;

Select Room\_Type, Avg(review\_scores\_rating) as Average\_Ratings from listing\_dallas\_df group by Room\_Type;

-iii-Bookings

Select room\_type, count(available) as No\_of\_Bookings from list\_availA

where available=0 group by room\_type;

Select room\_type, count(available) as No\_of\_Bookings from list\_availD

where available=0 group by room\_type;

-iv-Acceptance Rate

Select listD.Room\_Type, AVG(hostD.host\_acceptance\_rate) as Average\_Acceptance\_Rate from listing\_dallas\_df as listD

join host\_dallas\_df as hostD

ON listD.host\_id = hostD.host\_id

GROUP BY listD.Room\_Type;

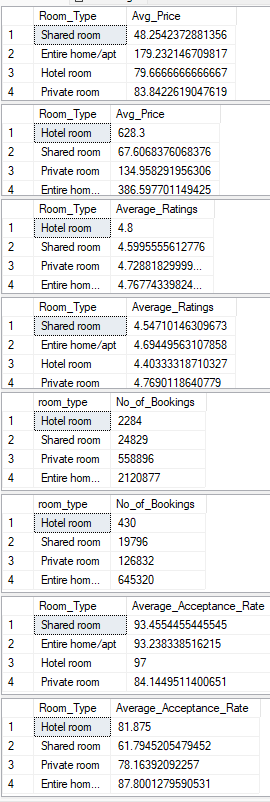
Select listA.Room\_Type, AVG(hostA.host\_acceptance\_rate) as Average\_Acceptance\_Rate from listing\_austin\_df as listA

join host\_austin\_df as hostA

ON listA.host\_id = hostA.host\_id

GROUP BY listA.Room\_Type;

OUTPUT:



RATIONALE:

* To compare Average price, Ratings, Bookings and Average acceptance rate across listing categories between two cities.