Data Scientist Role Play: Profiling and Analyzing the Yelp Dataset Coursera Worksheet

This is a 2-part assignment. In the first part, you are asked a series of questions that will help you profile and understand the data just like a data scientist would. For this first part of the assignment, you will be assessed both on the correctness of your findings, as well as the code you used to arrive at your answer. You will be graded on how easy your code is to read, so remember to use proper formatting and comments where necessary.

In the second part of the assignment, you are asked to come up with your own inferences and analysis of the data for a particular research question you want to answer. You will be required to prepare the dataset for the analysis you choose to do. As with the first part, you will be graded, in part, on how easy your code is to read, so use proper formatting and comments to illustrate and communicate your intent as required.

For both parts of this assignment, use this "worksheet." It provides all the questions you are being asked, and your job will be to transfer your answers and SQL coding where indicated into this worksheet so that your peers can review your work. You should be able to use any Text Editor (Windows Notepad, Apple TextEdit, Notepad ++, Sublime Text, etc.) to copy and paste your answers. If you are going to use Word or some other page layout application, just be careful to make sure your answers and code are lined appropriately.

In this case, you may want to save as a PDF to ensure your formatting remains intact for you reviewer.

Part 1: Yelp Dataset Profiling and Understanding

1. Profile the data by finding the total number of records for each of the tables below:

i. Attribute table = 10000

-- Code to get the total records from attribute table

select count(\*) as

total\_records

from attribute;

+---------------+

| total\_records |

+---------------+

| 10000 |

+---------------+

ii. Business table = 10000

-- Code to get the total records from business table

select count(\*) as

total\_records

from business;

iii. Category table = 10000

select count(\*) as

total\_records

from category;

iv. Checkin table = 10000

select count(\*) as

total\_records

from Checkin;

v. elite\_years table = 10000

select count(\*) as

total\_records

from elite\_years ;

vi. friend table = 10000

select count(\*) as

total\_records

from friend;

vii. hours table = 10000

select count(\*) as

total\_records

from hours;

viii. photo table = 10000

select count(\*) as

total\_records

from photo;

ix. review table = 10000

select count(\*) as

total\_records

from review;

x. tip table = 10000

select count(\*) as

total\_records

from tip;

xi. user table = 10000

select count(\*) as

total\_records

from user;

2. Find the total distinct records by either the foreign key or primary key for each table. If two foreign keys are listed in the table, please specify which foreign key.

i. Business = 10000

SELECT Count(distinct id)

FROM business;

+--------------------+

| Count(distinct id) |

+--------------------+

| 10000 |

|  |
| --- |
|  |

ii. Hours = 1562

SELECT Count(distinct business\_id)

FROM hours;

iii. Category = 2643

SELECT Count(distinct id)

FROM Category;

iv. Attribute = 1115

SELECT Count(distinct business\_id)

FROM attribute;

v. Review = (id)10000 , (business\_id)8090, (user\_id)9581

SELECT Count(distinct id)

FROM review

SELECT Count(distinct business\_id)

FROM Review;

SELECT Count(distinct user\_id)

FROM review;

vi. Checkin = 493

SELECT Count(distinct business\_id)

FROM Checkin;

vii. Photo = (distinct id)= 10000 , (business\_id) = 6493

SELECT Count(distinct id)

FROM photo;

SELECT Count(distinct business\_id)

FROM photo;

viii. Tip (user\_id) =537 , (business\_id) = 3979

SELECT Count(distinct user\_id)

FROM tip;

SELECT Count(distinct business\_id)

FROM tip;

ix. User (id) = 10000

SELECT Count(distinct id)

FROM user;

x. Friend (user\_id) = 11

SELECT Count(distinct user\_id)

FROM friend;

xi. Elite\_years (user\_id) = 2780

SELECT Count(distinct user\_id

FROM elite\_years;

Note: Primary Keys are denoted in the ER-Diagram with a yellow key icon.

3. Are there any columns with null values in the Users table? Indicate "yes," or "no."

Answer: No

SQL code used to arrive at answer:

SELECT \*

FROM user

WHERE id IS NULL OR

name IS NULL OR

review\_count IS NULL OR

yelping\_since IS NULL OR

useful IS NULL OR

cool IS NULL OR

fans IS NULL OR

average\_stars IS NULL OR

compliment\_hot IS NULL OR

compliment\_more IS NULL OR

compliment\_profile IS NULL OR

compliment\_cute IS NULL OR

compliment\_list IS NULL OR

compliment\_note IS NULL OR

compliment\_plain IS NULL OR

compliment\_cool IS NULL OR

compliment\_funny IS NULL OR

compliment\_writer IS NULL OR

compliment\_photos IS NULL

4. For each table and column listed below, display the smallest (minimum), largest (maximum), and average (mean) value for the following fields:

i. Table: Review, Column: Stars

min:1 max:5 avg:3.7082

SELECT min(stars)

FROM review;

SELECT max(stars)

FROM review;

SELECT avg(stars)

FROM review;

ii. Table: Business, Column: Stars

min:1 max:5 avg:3.6549

SELECT min(stars)

FROM business;

SELECT max(stars)

FROM business;

SELECT avg(stars)

FROM business;

iii. Table: Tip, Column: Likes

min:0 max: 2 avg: 0.0144

SELECT min(likes)

FROM tip;

SELECT max(likes)

FROM tip;

SELECT avg(likes)

FROM tip;

iv. Table: Checkin, Column: Count

min: 1 max:53 avg: 1.9414

SELECT min(count)

FROM checkin;

SELECT max(count)

FROM checkin;

SELECT avg(count)

FROM checkin;

v. Table: User, Column: Review\_count

min: 0 max:2000 avg: 24.2995

SELECT min(review\_count)

FROM user;

SELECT max(review\_count)

FROM user;

SELECT avg(review\_count)

FROM user;

5. List the cities with the most reviews in descending order:

--SQL code used to arrive at answer:

SELECT

city,

sum(review\_count) AS total\_review

FROM business

GROUP BY city

ORDER BY total\_review DESC;

--Copy and Paste the Result Below:

123456789

--

SELECT

city,

sum(review\_count) AS total\_review

FROM business

GROUP BY city

ORDER BY total\_review DESC;

Reset

+-----------------+--------------+

| city | total\_review |

+-----------------+--------------+

| Las Vegas | 82854 |

| Phoenix | 34503 |

| Toronto | 24113 |

| Scottsdale | 20614 |

| Charlotte | 12523 |

| Henderson | 10871 |

| Tempe | 10504 |

| Pittsburgh | 9798 |

| Montréal | 9448 |

| Chandler | 8112 |

| Mesa | 6875 |

| Gilbert | 6380 |

| Cleveland | 5593 |

| Madison | 5265 |

| Glendale | 4406 |

| Mississauga | 3814 |

| Edinburgh | 2792 |

| Peoria | 2624 |

| North Las Vegas | 2438 |

| Markham | 2352 |

| Champaign | 2029 |

| Stuttgart | 1849 |

| Surprise | 1520 |

| Lakewood | 1465 |

| Goodyear | 1155 |

+-----------------+--------------+

(Output limit exceeded, 25 of 362 total rows shown)

6. Find the distribution of star ratings to the business in the following cities:

i. Avon

SQL code used to arrive at answer:

SELECT

stars,

sum(review\_count) AS stars\_rating\_count

FROM business

WHERE city = "Avon"

GROUP BY stars;

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

+-------+--------------------+

| stars | stars\_rating\_count |

+-------+--------------------+

| 1.5 | 10 |

| 2.5 | 6 |

| 3.5 | 88 |

| 4.0 | 21 |

| 4.5 | 31 |

| 5.0 | 3 |

+-------+--------------------+

ii. Beachwood

SQL code used to arrive at answer:

SELECT

stars,

sum(review\_count) AS stars\_rating\_count

FROM business

WHERE city = "Beachwood"

GROUP BY stars;

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

+-------+--------------------+

| stars | stars\_rating\_count |

+-------+--------------------+

| 2.0 | 8 |

| 2.5 | 3 |

| 3.0 | 11 |

| 3.5 | 6 |

| 4.0 | 69 |

| 4.5 | 17 |

| 5.0 | 23 |

+-------+--------------------+

7. Find the top 3 users based on their total number of reviews:

SQL code used to arrive at answer:

SELECT

name,

sum(review\_count) AS review

FROM user

GROUP BY id

ORDER BY review DESC

LIMIT 3;

Copy and Paste the Result Below:

+--------+--------+

| name | review |

+--------+--------+

| Gerald | 2000 |

| Sara | 1629 |

| Yuri | 1339 |

+--------+--------+

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

Based on the analysis in the table below, we can see Gerald with a total of 2000 reviews but only 253 fans, that is an average of 7 fans per review. On the other hand, Eric has 1116 reviews but with only 16 fans; therefore, we can conclude that posing more reviews does not correlate with more fans.

SELECT

name,

sum(review\_count) AS total\_review,

fans,

sum(review\_count)/fans AS avg\_review\_per\_fan

FROM user

GROUP BY id

ORDER BY total\_review DESC;

Result Table:

+-----------+--------------+------+--------------------+

| name | total\_review | fans | avg\_review\_per\_fan |

+-----------+--------------+------+--------------------+

| Gerald | 2000 | 253 | 7 |

| Sara | 1629 | 50 | 32 |

| Yuri | 1339 | 76 | 17 |

| .Hon | 1246 | 101 | 12 |

| William | 1215 | 126 | 9 |

| Harald | 1153 | 311 | 3 |

| eric | 1116 | 16 | 69 |

| Roanna | 1039 | 104 | 9 |

| Mimi | 968 | 497 | 1 |

| Christine | 930 | 173 | 5 |

| Ed | 904 | 38 | 23 |

| Nicole | 864 | 43 | 20 |

| Fran | 862 | 124 | 6 |

| Mark | 861 | 115 | 7 |

| Christina | 842 | 85 | 9 |

| Dominic | 836 | 37 | 22 |

| Lissa | 834 | 120 | 6 |

| Lisa | 813 | 159 | 5 |

| Alison | 775 | 61 | 12 |

| Sui | 754 | 78 | 9 |

| Tim | 702 | 35 | 20 |

9. Are there more reviews with the word "love" or with the word "hate" in them?

Answer: Based on table analysis, there are more reviews with the word “love”.

SQL code used to arrive at answer:

SELECT

feelings,

count(\*) AS total\_count

FROM (SELECT

CASE WHEN text LIKE "%love%" THEN "love"

WHEN text LIKE "%hate%" THEN "hate"

ELSE "Others"

END feelings

FROM review)

GROUP BY feelings

ORDER BY total\_count DESC;

+----------+-------------+

| feelings | total\_count |

+----------+-------------+

| Others | 8042 |

| love | 1780 |

| hate | 178 |

+----------+-------------+

10. Find the top 10 users with the most fans:

SQL code used to arrive at answer:

SELECT

name,

sum(fans) AS total\_fans

FROM user

GROUP BY id

ORDER BY total\_fans DESC

LIMIT 10;

Copy and Paste the Result Below:

+-----------+------------+

| name | total\_fans |

+-----------+------------+

| Amy | 503 |

| Mimi | 497 |

| Harald | 311 |

| Gerald | 253 |

| Christine | 173 |

| Lisa | 159 |

| Cat | 133 |

| William | 126 |

| Fran | 124 |

| Lissa | 120 |

+-----------+------------+

Part 2: Inferences and Analysis

1. Pick one city and category of your choice and group the businesses in that city or category by their overall star rating. Compare the businesses with 2-3 stars to the businesses with 4-5 stars and answer the following questions. Include your code.

A. I choose "Charlotte" as city and "Shopping" as category.

SELECT CASE WHEN stars >= 4 THEN "4-5 stars"

WHEN stars >= 2 THEN "2-3 stars"

ELSE "below 2"

END star\_rank,

city,

c.category,

count(distinct business.id) AS company\_count,

count(h.hours) AS working\_days

FROM business

JOIN hours h ON business.id = h.business\_id

JOIN category c ON business.id = c.business\_id

WHERE city = " Charlotte " AND c.category = "Shopping"

GROUP BY star\_rank;

+-----------+-----------+----------+---------------+--------------+

| star\_rank | city | category | company\_count | working\_days |

+-----------+-----------+----------+---------------+--------------+

| 2-3 stars | Charlotte | Shopping | 2 | 11 |

| 4-5 stars | Charlotte | Shopping | 1 | 7 |

+-----------+-----------+----------+---------------+--------------+

i. Do the two groups you chose to analyze have a different distribution of hours?

Yes, 2-3 stars has a total of 11 working days and 4-5 stars has 7.

ii. Do the two groups you chose to analyze have a different number of reviews?

A. Yes, the total number of reviews from 2-3 stars is more than doubled compared to 4-5 stars.

--Code to get total\_reviews

SELECT CASE WHEN stars >= 4 THEN "4-5 stars"

WHEN stars >= 2 THEN "2-3 stars"

ELSE "below 2"

END star\_rank,

city,

c.category,

count(distinct business.id) AS company\_count,

sum(review\_count) AS total\_review

FROM business

JOIN category c ON business.id = c.business\_id

WHERE city = "Charlotte" AND c.category = "Shopping"

GROUP BY star\_rank;

+-----------+-----------+----------+---------------+--------------+

| star\_rank | city | category | company\_count | total\_review |

+-----------+-----------+----------+---------------+--------------+

| 2-3 stars | Charlotte | Shopping | 2 | 14 |

| 4-5 stars | Charlotte | Shopping | 1 | 5 |

+-----------+-----------+----------+---------------+--------------+

iii. Are you able to infer anything from the location data provided between these two groups? Explain.

A. Stores that have 2-3 stars are within the South area, whereas 4-5 stars stores are from University City.

SQL code used for analysis:

SELECT CASE WHEN stars >= 4 THEN "4-5 stars"

WHEN stars >= 2 THEN "2-3 stars"

ELSE "below 2"

END star\_rank,

address,

neighborhood,

city,

postal\_code

FROM business

JOIN category c ON business.id = c.business\_id

WHERE city = "Charlotte" AND c.category = "Shopping"

ORDER BY star\_rank;

Table

Description automatically generated

2. Group business based on the ones that are open and the ones that are closed. What differences can you find between the ones that are still open and the ones that are closed? List at least two differences and the SQL code you used to arrive at your answer.

i. Difference 1:

Total review is significantly higher between still open and closed businesses.

ii. Difference 2:

Average reviews for still open is higher than closed ones being 32 and 23 respectively.

SQL code used for analysis:

SELECT CASE WHEN is\_open = 1 THEN "STILL OPEN"

WHEN is\_open = 0 THEN "CLOSED"

END status,

count(distinct id) AS tot\_company,

sum(review\_count) AS total\_review,

round(avg(review\_count),2) AS avg\_review,

round(avg(stars),2) AS avg\_stars

FROM business

GROUP BY is\_open

ORDER BY status DESC;

+------------+-------------+--------------+------------+-----------+

| status | tot\_company | total\_review | avg\_review | avg\_stars |

+------------+-------------+--------------+------------+-----------+

| STILL OPEN | 8480 | 269300 | 32.0 | 4.0 |

| CLOSED | 1520 | 35261 | 23.0 | 4.0 |

+------------+-------------+--------------+------------+-----------+

3. For this last part of your analysis, you are going to choose the type of analysis you want to conduct on the Yelp dataset and are going to prepare the data for analysis.

Ideas for analysis include: Parsing out keywords and business attributes for sentiment analysis, clustering businesses to find commonalities or anomalies between them, predicting the overall star rating for a business, predicting the number of fans a user will have, and so on. These are just a few examples to get you started, so feel free to be creative and come up with your own problem you want to solve. Provide answers, in-line, to all of the following:

i. Indicate the type of analysis you chose to do:

I wanted to see people’s choice in picking foods from different countries and their reviews.

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:

I picked several cuisines from several countries: “Chinese”,”Mexican”,”Korean”,”French”,”Italian”,”Japanese” and

“Indian”. Then I analyzed their star ratings and number of

reviews so that I can get some insights on which type of food is

popular amongst people on yelp.

iii. Output of your finished dataset:

+----------+-----------------+---------------+-------------------+-----------+

| category | Num\_Restaurants | AVG(stars) | AVG(review\_count) | city |

+----------+-----------------+---------------+-------------------+-----------+

| Korean | 7 | 4.5 | 8.0 | Toronto |

| French | 12 | 4.0 | 135.083333333 | Las Vegas |

| Chinese | 13 | 3.76923076923 | 423.230769231 | Las Vegas |

| Mexican | 28 | 3.625 | 73.0 | Edinburgh |

| Italian | 13 | 3.53846153846 | 78.2307692308 | Montréal |

| Japanese | 20 | 3.475 | 22.85 | Toronto |

+----------+-----------------+---------------+-------------------+-----------+

iv. Provide the SQL code you used to create your final dataset:

SELECT c.category,COUNT(b.name) AS

Num\_Restaurants, AVG(stars), AVG(review\_count),b.city

FROM (business b INNER JOIN hours h ON b.id = h.business\_id)

INNER JOIN category c ON c.business\_id = b.id

WHERE c.category IN

("Chinese","Mexican","French","Italian","Korean","Japanese","Ind

ian")

GROUP BY c.category

ORDER BY AVG(stars) DESC;