FINAL EXAM

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:

```
SELECT *
FROM [table]
```

i. Attribute table = 10,000
ii. Business table = 10,000
iii. Category table = 10,000
iv. Checkin table = 10,000
v. elite_years table = 10,000
vi. friend table = 10,000
vii. hours table = 10,000
viii. photo table = 10,000
ix. review table = 10,000
x. tip table = 10,000
xi. user table = 10,000

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.

```
SELECT COUNT (
    DISTINCT [PK/FK])
FROM [table])
```

i. Business = id (PK - 10,000)
ii. Hours = business_id (FK - 1,562)
iii. Category = business_id (FK - 2,643)
iv. Attribute = business_id (FK - 1,115)
v. Review = id (PK - 10,000), business_id (FK - 8,090), user_id (FK - 9,581)
vi. Checkin = business_id (FK - 493)
vii. Photo = id (PK - 10,000), business_id (FK - 6,493)
viii. Tip = user_id (FK - 537), business_id (FK - 3,979)
ix. User = id (PK - 10,000)
x. Friend = user_id (FK - 11)
xi. Elite_years = user_id (FK - 2,780)

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 COUNT (*)
FROM user
WHERE [column] IS NULL
```

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

```
SELECT MIN([column])
,MAX([column])
,AVG([column])
FROM [table]
```

i. Table: Review, Column: Stars

min: 1 max: 5 avg: 3.7082

ii. Table: Business, Column: Stars

min: 1.0 max: 5.0 avg: 3.6549

iii. Table: Tip, Column: Likes

min: 0 max: 2 avg: 0.0144

iv. Table: Checkin, Column: Count

min: 1 max: 53 avg: 1.9414

v. Table: User, Column: Review_count

min: 0 max: 2000 avg: 24.2995

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

SQL code used to arrive at answer:

```
SELECT city AS City
,SUM(review_count) AS ReviewCount
FROM business
GROUP BY city
ORDER BY ReviewCount DESC
```

Copy and Paste the Result Below:

City	ReviewCount
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 city
,COUNT(id) AS 'Businesses'
,stars
FROM business
WHERE city LIKE 'Avon'
GROUP BY stars
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

city	Businesses	stars
Avon	1	1.5
Avon	2	2.5
Avon	3	3.5
Avon	2	4.0
Avon	1	4.5
Avon	1	5.0

ii. Beachwood

SQL code used to arrive at answer:

```
SELECT city
,COUNT(id) AS 'Businesses'
,stars
FROM business
WHERE city LIKE 'Beachwood'
GROUP BY stars
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

city	Businesses	stars
Beachwood	1	2.0
Beachwood	1	2.5
Beachwood	2	3.0
Beachwood	2	3.5
Beachwood	1	4.0
Beachwood	2	4.5
Beachwood	5	5.0

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

SQL code used to arrive at answer:

```
SELECT name AS Name
,review_count AS 'Review Count'
FROM user
GROUP BY id
ORDER BY review_count DESC
LIMIT 3
```

Copy and Paste the Result Below:

Name	Review Count
Gerald	2000
Sara	1629
Yuri	1339

8. Does posting more reviews correlate with more fans? Please explain your findings and interpretation of the results:

Not necessarily. A cursory examination of the data shows that Yelpers with
high review counts sometimes have fewer fans than Yelpers with lower review counts.

```
SELECT name AS Name
,review_count AS ReviewCount
,fans AS Fans
FROM user
ORDER BY review_count DESC
```

```
+-----------+-------------+------+
| Name      | ReviewCount | Fans |
+-----------+-------------+------+
| Gerald    |        2000 |  253 |
| Sara      |        1629 |   50 |
| Yuri      |        1339 |   76 |
| .Hon      |        1246 |  101 |
| William   |        1215 |  126 |
| Harald    |        1153 |  311 |
| eric      |        1116 |   16 |
| Roanna    |        1039 |  104 |
| Mimi      |         968 |  497 |
| Christine |         930 |  173 |
| Ed        |         904 |   38 |
| Nicole    |         864 |   43 |
| Fran      |         862 |  124 |
| Mark      |         861 |  115 |
| Christina |         842 |   85 |
| Dominic   |         836 |   37 |
| Lissa     |         834 |  120 |
| Lisa      |         813 |  159 |
| Alison    |         775 |   61 |
| Sui       |         754 |   78 |
| Tim       |         702 |   35 |
| L         |         696 |   10 |
| Angela    |         694 |  101 |
| Crissy    |         676 |   25 |
| Lyn       |         675 |   45 |
+-----------+-------------+------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

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

    Answer: Love

    SQL code used to arrive at answer:

```
SELECT
    (SELECT COUNT (*)
    FROM review
    WHERE text LIKE '%love%') AS LoveCount
    ,(SELECT COUNT (*)
    FROM review
    WHERE text LIKE '%hate%') AS HateCount
FROM review
LIMIT 1
```

```
+-----------+-----------+
| LoveCount | HateCount |
+-----------+-----------+
|      1780 |       232 |
+-----------+-----------+
```

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

    SQL code used to arrive at answer:

```
SELECT name
,fans
FROM user
GROUP BY id
ORDER BY fans DESC
```

```
LIMIT 10

    Copy and Paste the Result Below:

+-----------+------+
| name      | 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.

City: Toronto
Category: Restaurants


i. Do the two groups you chose to analyze have a different distribution of hours?
Four of the ten 'Restaurants' in Toronto do not have hours listed. Some open late morning and close before midnight, while others open in the evening and close after midnight.

SELECT b.city AS 'City'
,b.name AS 'Business'
,h.hours AS 'Hours'
,c.category AS 'Category'
FROM
    (business b INNER JOIN category c ON b.id = c.business_id)
    LEFT JOIN hours h ON b.id = h.business_id
WHERE b.city LIKE 'Toronto'
AND
c.category LIKE 'Restaurants'
GROUP BY b.id

+---------+--------------------+----------------------+-------------+
| City    | Business           |                Hours | Category    |
+---------+--------------------+----------------------+-------------+
| Toronto | Mama Mia           |                 None | Restaurants |
| Toronto | Cabin Fever        |  Saturday|16:00-2:00 | Restaurants |
| Toronto | Royal Dumpling     |                 None | Restaurants |
| Toronto | Big Smoke Burger   | Saturday|10:30-21:00 | Restaurants |
| Toronto | Sushi Osaka        | Saturday|11:00-23:00 | Restaurants |
| Toronto | Pizzaiolo          |  Saturday|10:00-4:00 | Restaurants |
| Toronto | 99 Cent Sushi      | Saturday|11:00-23:00 | Restaurants |
| Toronto | The Kosher Gourmet |                 None | Restaurants |
| Toronto | Naniwa-Taro        |                 None | Restaurants |
| Toronto | Edulis             | Saturday|18:00-23:00 | Restaurants |
+---------+--------------------+----------------------+-------------+


SELECT b.city AS 'City'
,COUNT(b.id) AS 'Business'
,h.hours AS 'Hours'
,c.category AS 'Category'
FROM
    (business b INNER JOIN category c ON b.id = c.business_id)
    LEFT JOIN hours h ON b.id = h.business_id
WHERE b.city LIKE 'Toronto'
AND
c.category LIKE 'Restaurants'
GROUP BY h.hours

+---------+----------+----------------------+-------------+
| City    | Business |                Hours | Category    |
+---------+----------+----------------------+-------------+
| Toronto |        4 |                 None | Restaurants |
```

```
| Toronto |        1 |   Friday|10:30-21:00 | Restaurants |
| Toronto |        2 |   Friday|11:00-23:00 | Restaurants |
| Toronto |        1 |   Friday|18:00-23:00 | Restaurants |
| Toronto |        1 |    Friday|18:00-2:00 | Restaurants |
| Toronto |        1 |     Friday|9:00-4:00 | Restaurants |
| Toronto |        1 |   Monday|10:30-21:00 | Restaurants |
| Toronto |        2 |   Monday|11:00-23:00 | Restaurants |
| Toronto |        1 |    Monday|16:00-2:00 | Restaurants |
| Toronto |        1 |    Monday|9:00-23:00 | Restaurants |
| Toronto |        1 |  Saturday|10:00-4:00 | Restaurants |
| Toronto |        1 | Saturday|10:30-21:00 | Restaurants |
| Toronto |        2 | Saturday|11:00-23:00 | Restaurants |
| Toronto |        1 |  Saturday|16:00-2:00 | Restaurants |
| Toronto |        1 | Saturday|18:00-23:00 | Restaurants |
| Toronto |        1 |   Sunday|10:00-23:00 | Restaurants |
| Toronto |        1 |   Sunday|11:00-19:00 | Restaurants |
| Toronto |        1 |   Sunday|11:00-23:00 | Restaurants |
| Toronto |        1 |   Sunday|12:00-16:00 | Restaurants |
| Toronto |        1 |   Sunday|14:00-23:00 | Restaurants |
| Toronto |        1 |    Sunday|16:00-2:00 | Restaurants |
| Toronto |        1 | Thursday|10:30-21:00 | Restaurants |
| Toronto |        2 | Thursday|11:00-23:00 | Restaurants |
| Toronto |        1 | Thursday|18:00-23:00 | Restaurants |
| Toronto |        1 |  Thursday|18:00-2:00 | Restaurants |
+---------+----------+----------------------+-------------+
(Output limit exceeded, 25 of 35 total rows shown)
```

ii. Do the two groups you chose to analyze have a different number of reviews?
Yes. Restaurants outside of Toronto appear to have a little more than double the average reviews than restaurants in Toronto (68.9 vs 29.9).

iii. Are you able to infer anything from the location data provided between these two groups? Explain.
Yes. 50% of Toronto restaurants have 4-5* reviews compared to 40% of restaurants outside Toronto. Therefore, you are more likely (though only slightly) to have a higher restaruant star rating in Toronto than outside Toronto.

SQL code used for analysis:

```
SELECT
    ((SELECT COUNT(b.stars)   --Counts # of Toronto rest. w/ 2*-3* reviews
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'
        AND
        b.stars BETWEEN 2.0 AND 3.0)
    ||'/'||
    (SELECT COUNT(b.stars)   --Counts # of Toronto rest. w/ 4*-5* reviews
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'
        AND
        b.stars BETWEEN 4.0 AND 5.0)
    ||'/'||
    (SELECT COUNT(b.stars)   --Counts total # of reviewed Toronto rest
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'))
    AS 'Toronto Rest. 2-3*/4-5*/All'
    ,(SELECT SUM(b.review_count)   --Counts total # of Toronto rest. reviews
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants')
    AS 'Toronto Rest. Total Reviews'
    ,((SELECT COUNT(b.stars)   --Counts # of all rest. outside Toronto w/ 2*-3* reviews
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city NOT LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'
        AND
        b.stars BETWEEN 2.0 AND 3.0)
    ||'/'||
    (SELECT COUNT(b.stars)   --Counts # of all rest. outside Toronto w/ 4*-5* reviews
        FROM business b INNER JOIN category c ON b.id = c.business_id
```

```
        WHERE b.city NOT LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'
        AND
        b.stars BETWEEN 4.0 AND 5.0)
    ||'/'||
    (SELECT COUNT(b.stars)   --Counts total # of all reviewed rest. outside Toronto
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city NOT LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants'))
    AS 'Rest. 2-3*/4-5*/All'
    ,(SELECT SUM(b.review_count)   --Counts total # of all rest. reviews outside Toronto
        FROM business b INNER JOIN category c ON b.id = c.business_id
        WHERE b.city NOT LIKE 'Toronto'
        AND
        c.category LIKE 'Restaurants')
    AS 'Rest. Total Reviews'
FROM business b INNER JOIN category c ON b.id = c.business_id
LIMIT 1
```

```
+----------------------------+----------------------------+--------------------+--------------------+
| Toronto Rest. 2-3*/4-5*/All | Toronto Rest. Total Reviews | Rest. 2-3*/4-5*/All | Rest. Total Reviews |
+----------------------------+----------------------------+--------------------+--------------------+
| 3/5/10                     |                        299 | 21/24/61           |               4205 |
+----------------------------+----------------------------+--------------------+--------------------+
```

2. Group businesses 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:
Closed businesses have a slightly lower average star rating than open businesses.

ii. Difference 2:
Open businesses have significantly more total reviews than closed businesses.

SQL code used for analysis:

```
SELECT
    ((SELECT COUNT(is_open)   --Counts # of closed businesses
        FROM business
        WHERE is_open = 0)
    ||' / '||
    (SELECT AVG(stars)   --Counts avg stars of closed businesses
        FROM business
        WHERE is_open = 0)
    ||' / '||
    (SELECT SUM(review_count)   --Counts review total of closed businesses
        FROM business
        WHERE is_open = 0))
    AS 'Closed / Average Stars / Total Reviews'
    ,((SELECT COUNT(is_open)   --Counts # of open businesses
        FROM business
        WHERE is_open = 1)
    ||' / '||
    (SELECT AVG(stars)   --Counts avg stars of open businesses
        FROM business
        WHERE is_open = 1)
    ||' / '||
    (SELECT SUM(review_count)   --Counts review total of open businesses
        FROM business
        WHERE is_open = 1))
    AS 'Open / Average Stars / Total Reviews'
FROM business
GROUP BY is_open
LIMIT 1
```

```
+--------------------------------------+-------------------------------------+
| Closed / Average Stars / Total Reviews | Open / Average Stars / Total Reviews |
+--------------------------------------+-------------------------------------+
| 1520 / 3.52039473684211 / 35261        | 8480 / 3.67900943396226 / 269300     |
+--------------------------------------+-------------------------------------+
```

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 decided to examine 'Nightlife' businesses and their success and star ratings by state.

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:
In order to examine 'Nightlife' businesses across all states I needed:
    1. Category
    2. State
    3. Average star rating
    4. Total number of reviews
    5. It's status (open or closed)
These data sets allow me to exmaine how many 'Nightlife' businesses were reviewed in each state, what average star rating and total number of reviews each business received, and whether said business is open or closed. By analyzing said data, I should be able to determine:
    1. Which state produces the most reviews of 'Nightlife' businesses
    2. The average star rating of 'Nightlife' businesses in each state
    3. The liklihood of a 'Nightlife' business succeeding or failing per state

iii. Output of your finished dataset:

Category	State	Business	Status	Average Stars	Total Reviews
Nightlife	AZ	Eklectic Pie - Mesa	Closed	4.0	129
Nightlife	AZ	Irish Republic	Closed	3.0	141
Nightlife	AZ	Innovative Vapors	Closed	4.5	11
Nightlife	AZ	Nabers Music, Bar & Eats	Closed	4.0	75
Nightlife	AZ	Gallagher's	Open	3.0	60
Nightlife	AZ	Bootleggers Modern American Smokehouse	Open	4.0	431
Nightlife	EDH	Mood	Closed	2.0	11
Nightlife	NV	Hi Scores - Blue Diamond	Open	3.5	105
Nightlife	OH	TWIISTED Burgers & Sushi	Open	4.0	94
Nightlife	OH	The Wine Mill	Open	4.5	42
Nightlife	OH	Brubaker's Pub	Open	3.0	5
Nightlife	OH	Cabin Club	Open	4.0	105
Nightlife	ON	Cabin Fever	Open	4.5	26
Nightlife	ON	The Charlotte Room	Closed	3.5	10
Nightlife	ON	The Erin Mills Pump & Patio	Open	3.0	27
Nightlife	ON	The Fox & Fiddle	Open	2.5	35
Nightlife	ON	Halo Brewery	Open	4.0	15
Nightlife	PA	Moondogs Pub	Open	3.5	7
Nightlife	PA	Iron City Grille	Closed	2.0	3
Nightlife	QC	Restaurant Rosalie	Closed	3.0	19

category	State	Open/Closed	Success Rate	Average Stars	Total Reviews
Nightlife	AZ	2/4	33%	3.75	847
Nightlife	EDH	0/1	0%	2.0	11
Nightlife	NV	1/0	100%	3.5	105
Nightlife	OH	4/0	100%	3.875	246
Nightlife	ON	4/1	80%	3.5	113
Nightlife	PA	1/1	50%	2.75	10
Nightlife	QC	0/1	0%	3.0	19

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

```
SELECT c.category AS 'Category'
,b.state AS 'State'
,b.name AS 'Business'
,CASE
    WHEN b.is_open = 0
    THEN 'Closed'
    WHEN b.is_open = 1
    THEN 'Open'
```

```
      END AS 'Status'
,AVG(b.stars) AS 'Average Stars'
,SUM(b.review_count) AS 'Total Reviews'
FROM business b INNER JOIN category c ON b.id = c.business_id
WHERE c.category LIKE 'Nightlife'
GROUP BY b.id
ORDER by b.state ASC


SELECT c.category AS 'Category'
,b.state AS 'State'
,(SUM(b.is_open))
||'/'||
(COUNT(b.is_open)-SUM(b.is_open))
    AS 'Open/Closed'
,(SUM(b.is_open)*100/(SELECT COUNT(b.is_open)))||'%'
    AS 'Success Rate'
,AVG(b.stars) AS 'Average Stars'
,SUM(b.review_count) AS 'Total Reviews'
FROM business b INNER JOIN category c ON b.id = c.business_id
WHERE c.category LIKE 'Nightlife'
GROUP BY b.state


FINAL EXAM
```