TAEHOON KIM
JOVANY FUNES
MELISSA NUBLA

DATABASE 302

05/09/2018

Executive Summary

This project focuses on the reliability of Yelp's user ratings or average number of stars for different establishments across the country. This looks specifically at the most important information like location, contacts, hours, menu, average stars/ratings, etc. This is then related to external factors like average income in a certain area. Instead of focusing on the internal factors that affect the ratings like the quality of food, service, and sanitary conditions, external factors such as the average income, and average ratings in a specific zip code area were considered.

Information had initially been collected through the yelp website as it provides a zip file of selected data, which contained enough information for this study. However, there had been initial issues with the database, such as difficulty extracting and running the files. A simple solution to run the files through python had initially worked, but was clearly not sustainable for the size of the database needed. Finally, another search for yelp data yielded Kaggle.com, where the same yelp data had been made available in a file that was uncorrupted. After this process, the database is then populated, cleaned, normalized, processed, and analyzed.

Some parameters that the group focused on were the relationship of the rates and the general locations, average stars in a certain area, number of restaurants near a location, average income in the area, and the like. Other things observed in the study were subjects such as the polarity of reviews in a certain area. This looks at the percentage of positive to negative ratings for a given business.

This group utilized a few different data tools and/or languages to run the study. These were: Python for general data collection and some polarity tests. SQL, Excel and Access, for the general management of the database, from cleaning up, data queries, and joining information. The group also used Insights for some simple charting and data visualization, and finally, Caspio, a website used to create a UI for the database of this project.

The group has created a page on ArcGIS using the Insights application to demonstrate the data that the group has found. The link can be found here. (Because of a glitch, some of the language on the page is in Korean. This can be fixed by using a simple translator on your browser, or reaching out to one of the researchers.) The group has also created a simple website to provide a UI to the audience. It can be accessed through this link or the address: www.coloreddata.com

Problem Description

Yelp is the company that provides people information about what local businesses are located in which specific area and how people think of the business by showing the star out of 5. However, we thought that since the stars are reflected on how people think, it can be very subjective. For example, when two people give 3 stars, it may be possible that the stars mean differently. One might say that 3 stars means 'tasty' to him/her when other say that it indicates 'just okay'. Therefore, we realized that it would be better if there are many factors that supplement the drawback of star systems so that users can have more objective review on the local businesses. We found that on Yelp, it does not show the average stars in that local area. Our design shows not only the average star of its business, but also the overall average stars of businesses in the location so that users can compare how high the business is rated in the area.

Also, we assume that as people have experienced fancier and more expensive foods, they tend to be stricter on giving stars; people who are financially prosperous have the experience. Therefore, we added ZIP_INCOME table so that people can consider the same amount of stars differently by locations. For example, users compare the average income by city and if one's mean is higher than others', people can think that rates on restaurants in one city may be more strictly measured than in the other cities.

Furthermore, we show the review count of each business and the number of polarities of review texts on each business whether they are positive or negative. As the number of review is higher, the rate become more reliable since then the review is reflected by many people. Moreover, because the user can see the polarity, they can more accurately assume the meaning of stars without reading the whole text. For instance, if someone gave 3 stars with negative polarity text, the user can think that the 3 star is more like negative sign for the reviewer, not neutral.

We found that McDonald's, Subway, and Taco Bell are the three businesses that are the most widely located in our BUSINESS table. Also, we thought that they are the restaurants that provide the same quality of food in every stores. Therefore, we consider them as a good indicator to find out how strict people are on a specific area to give the stars like Big Mac index in economy. Unfortunately, those businesses do not cover every city we have in the dataset, but can be a helpful supplementary for star systems.

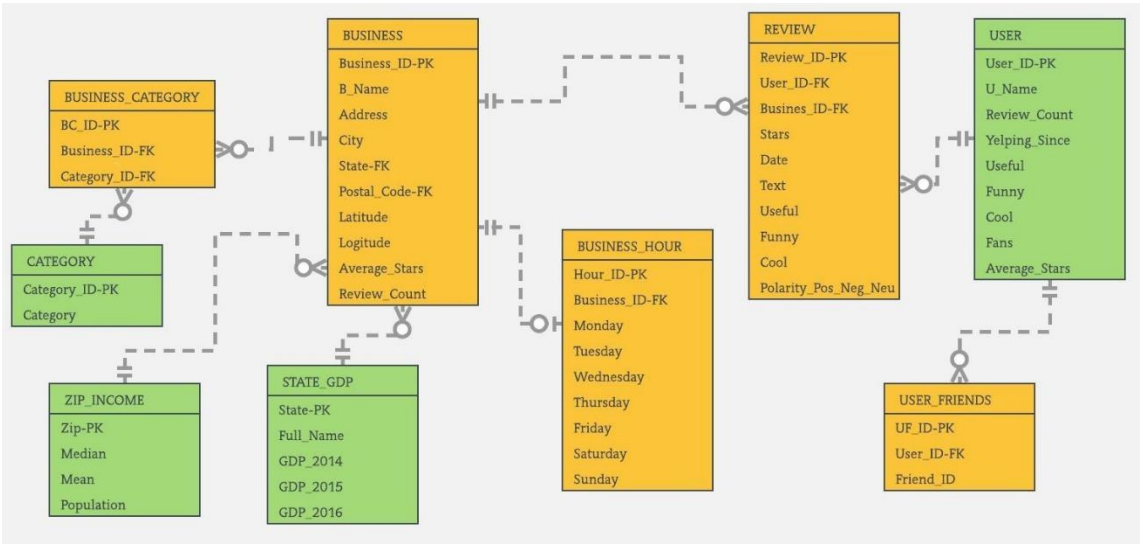
Database Design

Business Rules

- If BUSINESS or USER row is deleted, a row of REVIEW that is connected to those tables is also deleted.

- Any businesses that are not located in state that is in STATE_GDP or in postal code that is in ZIP_INCOME cannot be inserted on BUSINESS table.
- If CATEGORY row is to be deleted and that row is connected to any BUSINESS_CATEGORY row, the CATRGORY row deletion will not be allowed.

ER-Diagram



Normalization Process

➔ Started with four tables, BUSIENSS, BUSINESS_HOUR, REVIEW, and USER

➔ Found the violation of 1NF in Both BUSINESS and USER table.
(More than one element is located in one cell)

Business_ID	Categories
__8j8yhsmE98wNWHJNyAgw	Restaurants;Japanese;Sushi Bars
__aKnGBedQ51_hEc3D9ARw	Sandwiches;Restaurants;Food;Bakeries;Juice Bars & Smoothies
__bqGGnOjtY9eEhrZAUsgA	Thai;Restaurants
__eb2f_wEBrEl0xCyLqDeQ	Mexican;Restaurants
__fMLrmv9M1_W4kBvR2VnQ	Restaurants;Food;Fast Food;Ice Cream & Frozen Yogurt
__fyRzU8kL6HkVV3wgxfmQ	Barbeque;Nightlife;Bars;Music Venues;Arts & Entertainment;Cabaret;Restaur
__H_61gpm7eViPMbWxPZSg	Restaurants;Sandwiches;Fast Food

user_id	friends
JJ-aSuM4pCFPdkfoZ34q0Q	0njfJmB-7n84DlIgUByCNw, rFn3Xe3RqHxRSxWOU19Gpg, HVUAmApa0
h5ERTYn2vQ1QbjTZvfWPaA	jYiZnueCr7gVq9T34xoa7g, yFLXGdY6rpHt7hRiwEFMag, -lMOmerlmuxO
jYnkJR3T8yCERXywoVhWYA	hkXekeW_Jj6mly8r8N7r1Q, dQDpV-VUtwYGqHznuRV-yw, Ez7Hve1SaSr
fV8Yr0c5tFQTQ2SRRJHXHw	HDb4fBWIAQ-foS8qLJty9w, x0hBZsmBTYxhjjx0MShz1A, aHKuHxlRDouN
aw973Pm1nrTbRjP4zY9B9g	Cit5yho-DqotA0BnXHErTQ, bm2DqfP4P454FjEtCbZdkQ, xruy2cNpgz-x9
pzpbr9mlagHhDRdin8DvPQ	PZbPhdy0_08tHprlJiZ4uw, i4dhajw93ZDmIa89n6-w9w, KyGnYa6biPYN0

➔ The attributes, categories and friends, would have multivalued dependency if they were put in each cell. Therefore, our group created new tables and put each element in a cell in the newly created tables.

CATEGORY with a primary key, Category_ID

CATEGORY

Category_ID	Category
C177	Arabian
C122	Bagels
C4	Bakeries
C24	Barbeque

BUSIENSS_CATEGORY (Intersection Table) with a primary key, BC_ID

⇨ Link the BUSINESS and CATEGORY table

⇨ Two foreign keys: Business_ID and Category_ID

BUSINESS_CATEGORY

BC_ID	Business_ID	Category_ID
BF1	__aKnGBedQ51_hEc3D9ARw	C1
BF2	__aKnGBedQ51_hEc3D9ARw	C2
BF3	__aKnGBedQ51_hEc3D9ARw	C3

USER_FRIEND with a primary key, UF_ID

⇨ Foreign key: User_ID

USER_FRIEND

UF_ID	User_ID	Friend_ID
UF1	__G_KecuUwFgqo3DB-B1JQ	4wzS9SJLQB1IOF3U9fqFWA
UF2	__G_KecuUwFgqo3DB-B1JQ	2grYhtBiVU_I3GJfengurQ
UF3	__G_KecuUwFgqo3DB-B1JQ	KxwBQDmho7DkgXabUbobvA
UF4	__G_KecuUwFgqo3DB-B1JQ	p_W8KHw2WuM5Qg45-pVgHg

Addition of new tables and an attribute

➔ In order to supplement the star systems, we needed to add more data in our database.

➔ Decided to add two more tables related to income and one attribute in REVIEW table.

ZIP_INCOME with a primary key, Zip

⇨ The primary key is placed in BUSINESS table as a foreign key. It is called Postal_Code in BUSINESS.

ZIP_INCOME

Zip	Median	Mean	Pop
40311	41001	46706	7153
40312	31759	40888	5898
40313	27558	40340	3143
40322	18602	23054	1817
40324	62459	70884	41342
40328	36384	42673	1065
40330	45654	55149	19334

STATE_GDP with a primary key, State

⇨ The primary key is placed in BUSINESS table as a foreign key.

STATE_GDP

State	Full_Name	GDP_2014	GDP_2015	GDP_2016
AK	Alaska	58196	53311	50404
AL	Alabama	193995	200421	205625
AR	Arkansas	118068	119320	121383
AZ	Arizona	281031	292972	305849
CA	California	2358811	2505853	2622731

New Attribute in REVIEW

⇨ A new attribute, Polarity_Pos_Neg_Neu, is inserted.

REVIEW

Review_ID	Use	Busi	Star	Date	Text	Use	Fun	Co	Polarity_Pos_Neg_Neu
__ph3z9F6l7	1EMRJ	jyZqml	5	######	FRIED SA	2	0	0	Positive
__2vz3RDm7	NX95(	nvS8Q	1	######	Super di	0	0	0	Negative
__4wwyHKGX	K67NC	Mh2sy	4	######	I I love t	0	0	0	Positive
__559hm8on	h8k43	9TCzxl	4	######	Kids mer	1	0	0	Positive
__7F4FKm3eJ	2uZ8u	mKf7p	4	######	I don't k	3	7	3	Positive

Data Model

CATEGORY ----- BUSINESS_CATEGORY (1:M)

BUSINESS ----- BUSINESS_CATEGORY (1:M)

➔ BUSINESS and CATEGORY tables have a Many to Many (N:M) relationship. One BUSINESS can have many categories and one CATEGORY can be involved in many businesses. Therefore, we made an intersection table, BUSINESS_CATEGORY. The table has one row for each line between BUSINESS and CATEGORY and each of two tables gets 1:M relationship with the intersection table. Originally, the intersection table has the combination of parents keys (Primary keys of BUSINESS and CATEGORY) as primary key. However, we made its own primary key, BC_ID, since we did not want the composite key, so the parent keys, Business_ID and Category_ID, became foreign keys in the REVIEW table.

BUSINESS ----- REVIEW (1:M)

USER ----- REVIEW (1:M)

➔ BUSINESS and USER each has 1:M relationship with REVIEW: one BUSINESS or one USER can have many reviews, but one REVIEW can only have one BUSINESS and USER. Therefore, BUSINESS and USER are the parent sides while REVIEW is child side, so primary keys of BUSINESS and USER, Business_ID and User_ID, become foreign key when they are linked to REVIEW.

ZIP_INCOME ----- BUSINESS (1:M)

STATE_GDP ----- BUSINESS(1:M)

➔ In our model, every BUSINESS is enforced to be in one of postal code in ZIP_INCOME and states in STATE_GDP. Therefore, BUSINESS have 1:M relationship with those two tables: one ZIP or STATE can have many businesses, but one BUSIENSS can have only one zip and state. Since they have 1:M relationship, primary keys of ZIP_INCOME and STATE_GDP, Zip and State, are placed in BUSINESS when the tables are linked.

BUSINESS ----- BUSINESS_HOUR (1:1)

➔ One BUSINESS have their own open hour, so BUSINESS and BUSINESS_HOUR have a 1:1 relationship. Primary key of BUSINESS, Business_ID, is placed in BUSINESS_HOUR as a foreign key when they are linked together. Since sometimes BUSINESS does not provide its open hour, minimum cardinality is zero to BUSINESS_HOUR.

USER ----- USER_FRIENDS (1:M)

➔ USER_FRIEND is an intersection table between USER and FRIEND, which have M to N relationship, but we did not include FRIEND table since we would not use it. Therefore, USER and USER_FRIEND have 1:M relationship. Also, friend can be logically existed only when user exist, so minimum cardinality of USER is one while it is zero to USER_FRIEND. Primary key of USER, User_ID, is placed in USER_FRIEND as a foreign key when they are linked together.

System Documentation

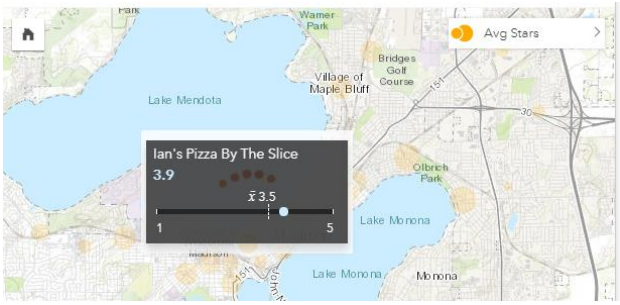
1. **SELECT** BU.City, BU.Business_ID, [B_Name], Review_Count, Avg_Star_Business, Avg_Star_City
   FROM (SELECT B.City, RS.Business_ID, [B_Name], Review_Count, Avg_Star_Business
   FROM BUSINESS AS B,
   (SELECT Business_ID, count(Business_ID) AS Review_Count,
   Round(Sum(Stars)/count(Business_ID), 1) AS Avg_Star_Business
   FROM REVIEW GROUP BY Business_ID) AS RS
   WHERE RS.Business_ID = B.Business_ID) AS BU
   RIGHT OUTER JOIN
   (SELECT City, ROUND(AVG(Stars),1) AS AVG_Star_City
   FROM REVIEW AS R, BUSINESS AS B
   WHERE R.Business_ID = B.Business_ID GROUP BY City) AS CI

ON BU.City = CI.City;

➔ It enables people to understand how high the business is rated in its city by comparing its own star and average star of whole businesses in the city. Also, by looking at the review count, users are able to know that how many people's opinions are reflected on the stars. Therefore, it shows the level of objectivity.

AVG_STAR_BUSINESS_CITY

City	Business_ID	B_Name	Review_Count	Avg_Star_Business	Avg_Star_City
Ahwatukee	eVv6cnwhabK8ig5Di	Florencia Pizza Bistro	2	5	3.5
Ahwatukee	0Rni7ocMC_Lg2UH0l	Barro's Pizza	3	2.3	3.5
Ahwatukee	qJf61TR4Jq9Xph5RiX	Cupz N' Crepes	3	3.7	3.5
Allison Park	oQKOijnFplXhmobb9	Stak Shak Table and Taps	1	2	3.3
Allison Park	wuEJ2XIM97g9ilhDR(	K Asian Bistro	4	3.5	3.3
Allison Park	X7BA_EijhAWYCNuM	Kanlaya Thai Kitchen	1	5	3.3
Allison Park	QQIwwU1tyOyKwr_s(	Wild Sage	1	4	3.3
Allison Park	BnMnAotMjCw7mYlY	The Tuscan Inn	2	2.5	3.3
Ambridge	zBeUDwWx73QTZ34	Vocelli Pizza	1	4	4.6
Ambridge	2_7lYF6P2cYKnSiPhh	Orbit Inn	1	4	4.6
Ambridge	729grSa1Wsn-hfv7D	Pizza House	4	5	4.6



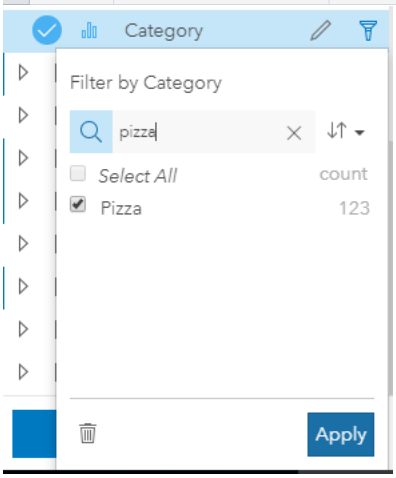
B_Name	COUNT of BUSINESS - REVIEW - USER - B...
Glass Nickel Pizza	11
Pizza Brutta	11
Ian's Pizza By The Slice	9
Ale Asylum	8
Natt Spil	8
Saigon Noodles	7
Paisan's	6
Sujeo	6
Noodles & Company	5
Pizza Di Roma	5

2. **SELECT** DISTINCT State, City, Postal_Code, B_Name, Category
   FROM BUSINESS AS B, BUSINESS_CATEGORY AS BC, CATEGORY AS C
   WHERE B.Business_ID = BC.Business_ID and BC.Category_ID = C.Category_ID and
   C.Category LIKE '*Pizza*'
   ORDER BY State, City;

➔ It shows every pizza stores in our database. By adding the state or city and changing the category in WHERE condition, people are able to search specific kinds of restaurant in a specific location.

BUSINESS_BY_CAT

State	City	Postal_Code	B_Name	Category
AZ	Ahwatukee	85044	Barro's Pizza	Pizza
AZ	Ahwatukee	85044	Florencia Pizza Bistro	Pizza
AZ	Anthem	85086	Barro's Pizza	Pizza
AZ	Avondale	85323	1 Brothers Pizza	Pizza
AZ	Avondale	85323	99 Pub and Grill	Pizza
AZ	Avondale	85323	Domino's Pizza	Pizza
AZ	Avondale	85323	Hungry Howie's Pizza	Pizza
AZ	Avondale	85323	Pizza Hut	Pizza

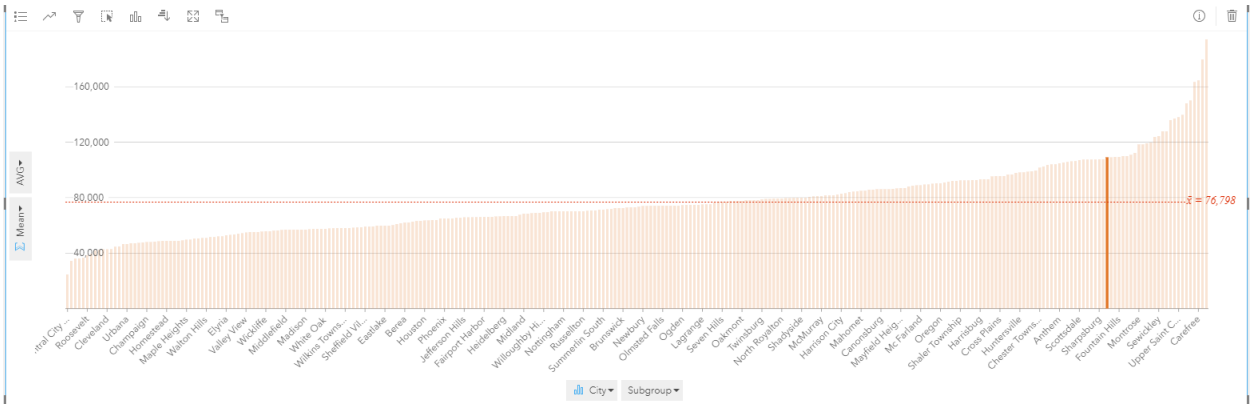


3. **SELECT** *
   FROM (SELECT State, City, Count(City) AS NumOfRes_In_City, ROUND(AVG(Mean),2)
   AS Avg_Mean_City_Income
   FROM BUSINESS AS B, ZIP_INCOME AS Z
   WHERE B.Postal_Code = Z.Zip
   GROUP BY City, State
   ORDER BY State, City) AS A;

➔ It shows the average income of each city. Our group assumes that people with higher income would be stricter on giving stars. Therefore, it would be one of good indicators to understand about the rate objectively. Higher the income of a city, stricter people in the city are to give star.

CITY_MEAN_VALUE

State	City	NumOfRes_In_City	Avg_Mean_City_Income
AZ	Avondale	98	70558.83
AZ	Carefree	13	164617
AZ	Cave Creek	57	119386
AZ	Chandler	509	85049.1
AZ	El Mirage	11	54394
AZ	Fountain Hills	45	109617.87
AZ	Gilbert	327	95761.92

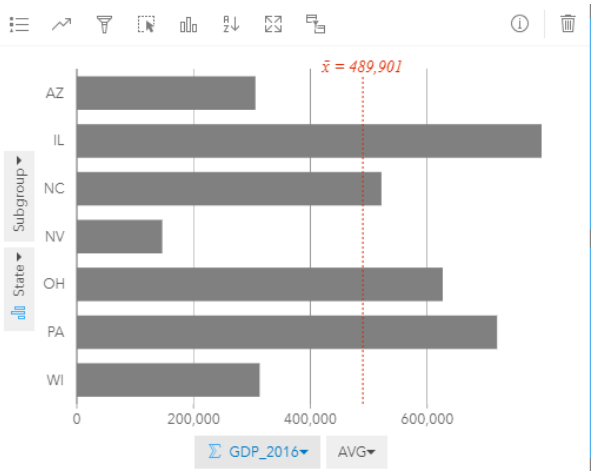


4. **SELECT** State, ROUND(GDP_2016, 2) AS AVG_GDP
   FROM STATE_GDP
   WHERE State IN (SELECT DISTINCT State
   FROM BUSINESS);

➔ Like #3 above, our group thought that average GDP can be another good indicator to let people understand the true meaning of the stars. Higher the GDP of state, stricter people in the state are to give star.

AVG_GDP_BY_STATE

State	AVG_GDP
NV	146278
AZ	305849
WI	313088
NC	521621
OH	626622
PA	719834
IL	796012



5. **SELECT** S5.state, STAR_1, STAR_2, STAR_3, STAR_4, STAR_5
FROM (SELECT S123.state, STAR_1, STAR_2, STAR_3, STAR_4
FROM (SELECT S12.state, STAR_1, STAR_2, STAR_3
FROM (SELECT S1.state, STAR_1, STAR_2
FROM (SELECT state, count(state) AS STAR_1
FROM Review AS R, Business AS B
WHERE R.business_id = B.business_id and R.stars = 1
GROUP BY state) AS S1,
(SELECT state, count(state) AS STAR_2
FROM Review AS R, Business AS B
WHERE R.business_id = B.business_id and R.stars = 2
GROUP BY state) AS S2
WHERE S1.state = S2.state) AS S12,
(SELECT state, count(state) AS STAR_3
FROM Review AS R, Business AS B
WHERE R.business_id = B.business_id and R.stars = 3
GROUP BY state) AS S3
WHERE S12.state = S3.state) AS S123,
(SELECT state, count(state) AS STAR_4
FROM Review AS R, Business AS B
WHERE R.business_id = B.business_id and R.stars = 4
GROUP BY state) AS S4
WHERE S123.state = S4.state) AS S1234,
(SELECT state, count(state) AS STAR_5
FROM Review AS R, Business AS B
WHERE R.business_id = B.business_id and R.stars = 5
GROUP BY state) AS S5
WHERE S1234.state = S5.state;

➔ It indicates how people in one's state tend to give stars. For example, we can tell people in WI tend to give a higher star than people in IL.

NumOfStars_BY_STATE

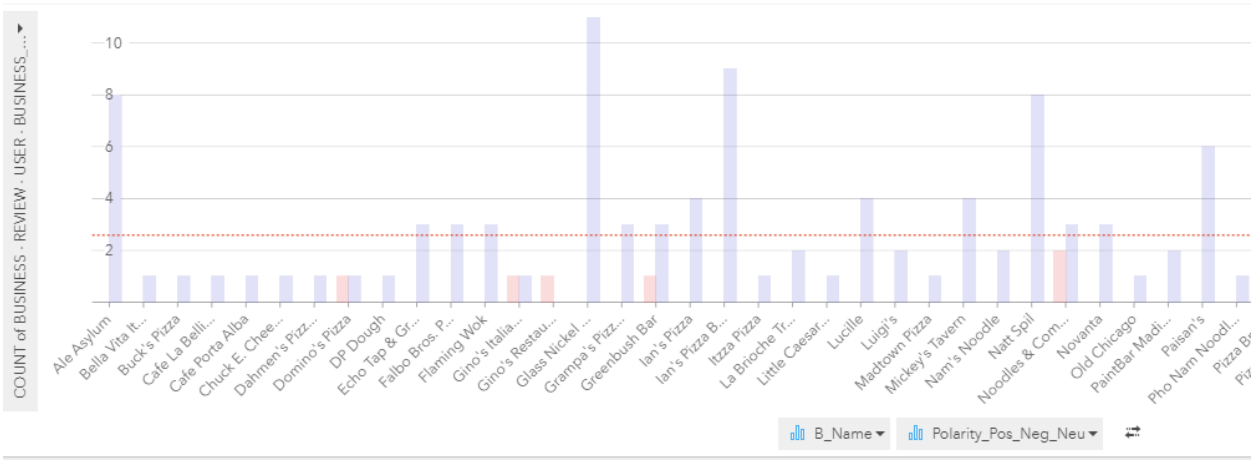
state	STAR_1	STAR_2	STAR_3	STAR_4	STAR_5
AZ	3017	2353	2997	6278	10781
IL	98	61	97	167	204
NC	653	612	769	1714	2040
NV	3202	2551	3531	7445	11525
OH	531	472	662	1275	1708
PA	444	424	498	1070	1409
WI	191	210	345	710	767

6. **SELECT** N.State, N.City, N.B_Name, Count_Positive, Count_Negative
FROM (SELECT State, City, B_Name, R.Business_ID, COUNT(R.Business_ID) AS Count_Positive
FROM BUSINESS AS B, REVIEW AS R
WHERE B.Business_ID = R.Business_ID and Polarity_Pos_Neg_Neu = 'Positive'
GROUP BY R.Business_ID,B_Name, City, State) AS P
RIGHT OUTER JOIN
(SELECT State, City, B_Name, R.Business_ID, COUNT(R.Business_ID) AS Count_Negative
FROM BUSINESS AS B, REVIEW AS R
WHERE B.Business_ID = R.Business_ID and Polarity_Pos_Neg_Neu = 'Negative'
GROUP BY R.Business_ID,B_Name, City, State) AS N
ON P.Business_ID = N.Business_ID
UNION
SELECT P.State, P.City, P.B_Name, Count_Positive, Count_Negative
FROM (SELECT State, City, B_Name, R.Business_ID, COUNT(R.Business_ID) AS Count_Positive
FROM BUSINESS AS B, REVIEW AS R
WHERE B.Business_ID = R.Business_ID and Polarity_Pos_Neg_Neu = 'Positive'
GROUP BY R.Business_ID,B_Name, City, State) AS P
LEFT OUTER JOIN
(SELECT State, City, B_Name, R.Business_ID, COUNT(R.Business_ID) AS Count_Negative
FROM BUSINESS AS B, REVIEW AS R
WHERE B.Business_ID = R.Business_ID and Polarity_Pos_Neg_Neu = 'Negative'
GROUP BY R.Business_ID,B_Name, City, State) AS N
ON P.Business_ID = N.Business_ID;

➔ It shows the polarity of review text of each business in REVIEW table. By looking at the chart below, people can know what the rate means more accurately. For example, let's say that there are two restaurants with average 3 stars: one with only positive polarity reviews and the other one with some positive and some negative reviews. Now user can understand those 3stars are not equivalent.

COUNT_POLARITY

State	City	B_Name	Count_Positive	Count_Negative
NV	Henderson	Hibachi-San		1
NV	Henderson	Hi-Coffee Cafe	1	1
NV	Henderson	Hiroba Sushi	8	
NV	Henderson	Hometown Classic Amer	6	
NV	Henderson	HoneyBaked Ham	2	
NV	Henderson	Hooters		1
NV	Henderson	Hooters Eastern	2	
NV	Henderson	Hot Dog Heaven	2	1
NV	Henderson	Hot Dog On A Stick	1	
NV	Henderson	Hot Head Burritos	3	1
NV	Henderson	Hot Rod Grille		1
NV	Henderson	HUMMUS	4	
NV	Henderson	I Love Sushi	26	2



7. **SELECT** State, City, B.business_id, B_name, R.stars
   FROM [User] AS U, Review AS R, Business AS B
   WHERE U.user_id = R.user_id and R.business_id = B.business_id
   and U.review_count >1000 and U.useful > 100
   ORDER BY State, City;

➔ Many review count and useful mark can be a good supplementary in order to understand the true meaning of the stars. The table below shows only reviews that have more than 1000 review count and more than 100 useful mark.

RELIABLE_USER_REVIEW

State	City	business_id	B_name	stars
AZ	Cave Creek	vz8HFGsITt6aj	El Encanto	3
AZ	Chandler	0xyqLvtr0Zbr4	Fleming's Prin	4
AZ	Gilbert	-6tvduBzjLI1IS	Joyride Taco H	4
AZ	Gilbert	BncSpY3IPix5s	Blue Wasabi S	5
AZ	Gilbert	wZukjLaf1V2d	Postino East	5
AZ	Glendale	1-EjdZhVZFuN	Popo's Fiesta	2

8. **SELECT** State, count(state) AS Num_Of_Restaurant
   FROM Business
   GROUP BY State;

➔ It shows how many restaurants in each state are in our database.

COUNT_RES_BY_STATE

State	Num_Of_Restaurant
AZ	6182
IL	283
NC	1930
NV	4404
OH	2020
PA	1507
WI	826

9. **SELECT** B_Name, AVG_Star
   FROM BUSINESS AS B,
   (SELECT Business_ID, AVG(Stars) AS AVG_Star
   FROM REVIEW GROUP BY Business_ID) AS R
   WHERE B.Business_ID = R.Business_ID and
   AVG_Star = 5 and State = 'NV' and City = 'Las Vegas';

➔ This one shows all the 5 star restaurants in a specific city.

HigestRate_Restaurant

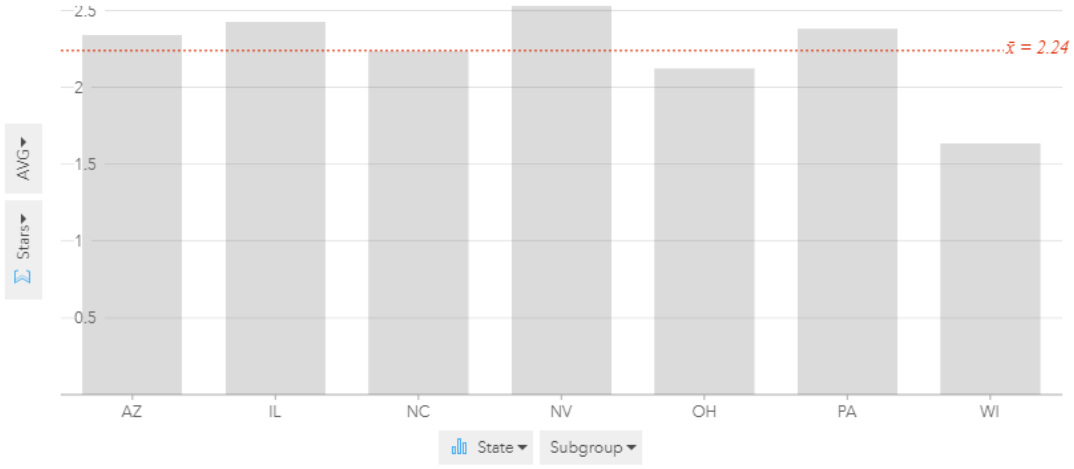
B_Name	AVG_Star
Jersey Mike's Subs	5
Nekter Juice Bar	5
Paid In Full	5
Chicken Itza	5
Ohana Hawaiian BBQ	5
Globe Cafe	5

10. **SELECT** State, ROUND(AVG(Stars),1) AS AVG_Star
    FROM BUSINESS AS B, REVIEW AS R
    WHERE B.Business_ID = R.Business_ID and
    B_Name IN ("McDonald's", "Subway", "Taco Bell")
    GROUP BY State;

➔ We figured out that "McDonald's", "Subway", and "Taco Bell" are the three restaurants that are the most widely distributed ones in our dataset. Also, our group assumed that they are the cheapest and the easiest ones, which people have a low preference. Therefore, we thought that the chart we made below can be like Big mac index in economy. By looking at the average stars of three restaurants in each state, people can have an idea how people in one location tend to give stars. For example, we could say that people in WI are the pickiest ones since the average star in WI in the chart below is the lowest.

FastFood_Index

State	AVG_Star
AZ	2.3
IL	2.4
NC	2.2
NV	2.5
OH	2.1
PA	2.4
WI	1.6



Database Administration

The easiest way to do this would be to have the YELP company have their own data server and save the database, as it might be easier to be able to connect the external data with what they already have. This might also allow them to have more security and protection in their process as they have a little more control over the database.

Additionally, it should probably be password protected. Otherwise, business owner might be tempted to change the rates and comments on its business and users would receive wrong information. Therefore, the log-in page(authentication) should be made and DBA grant the rights, following the user's roles.

DATABASE RIGHTS GRANTED				
Table	User(Yelper)	Business Owner	Management	System Administrator
USER	Read, Insert, Change, Delete	Read	Read, Insert, Change	Grant Rights, Modify Structure

USER_FRIENDS	Read, Insert, Change, Delete	Read	Read, Insert, Change	Grant Rights, Modify Structure
REVIEW	Read, Insert, Change, Delete	Read	Read, Insert, Change, Delete	Grant Rights, Modify Structure
BUSINESS	Read	Read, Insert, Change, Delete	Read, Insert, Change	Grant Rights, Modify Structure
BUSINESS_CATEGORY	Read	Read, Insert, Change, Delete	Read, Insert, Change, Delete	Grant Rights, Modify Structure
CATEGORY	Read	Read, Insert	Read, Insert, Change, Delete	Grant Rights, Modify Structure
BUSINESS_HOUR	Read	Read, Insert, Change, Delete	Read, Insert, Change, Delete	Grant Rights, Modify Structure
ZIP_INCOME	Read	Read	Read, Insert, Change, Delete	Grant Rights, Modify Structure
STATE_GDP	Read	Read	Read, Insert, Change, Delete	Grant Rights, Modify Structure

Yelp will mainly be collecting data from users and business owners. Business owners will leave the information that is in BUSINESS and BUSINESS_HOUR table and Users will provide the attributes of USER table. Also, User will write the comments on business and rate them with stars. When user gives the stars in the REVIEW table, the attributes of BUSINESS, Average_Stars and Review_Count, will automatically be recalculated by the amount.

Therefore, those two attributes of the business should be locked with shared lock. Other users can view the average stars and review count of the business, but cannot change at that moment by inserting row of REVIEW. Otherwise, they might be able to have concurrent update problem. The lock would be optimistic since there will not be many cases when two or more people provide review to the same business at the same time.

This might be helpful to the yelp company itself as we live in an age where knowledge equals power. This means that the more information yelp is able to give out about the businesses, it may attract more users and additionally, traffic as it would be much more useable to a larger range of people. For example, it adds a certain element of customization for the user as it gives them more choices or factors to add into their search for their perfect establishment. Next, it could be helpful for businesses marketing aspect as it will know more about which audience to focus on, what sells, what markets are available. This goes similarly for establishment owners as it gives them much more room to find more and different avenues to focus on improving their businesses.

Some future suggestions could be to create a more sustainable system that can be automatically updated, and involve much more data. This database uses only a sample set provided by yelp but having this run on a larger scale, not only for yelp, but also different companies that use an interface similar to that of yelp's can possibly be a good business move. Aside from the different stakeholders mentioned earlier, it might also be a great venture for yelp to look into the research business as it generates such a great set of data, especially considering the number of users and reviews in gets per day.