Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

ECE 656 Winter Project Report

Abstract

In the internet world, web service providers have the capability to collect and maintain large quantities of data, from their internet users. This data can be further processed and analyzed to better understand and also predict the requirements of their users. These service providers might be interested in targeting specific users, or groups of classified users (e.g. based on their geographic area, age, education, etc.), as a part of their business strategy. Prior to processing large quantities of data, the data would should be made ready for processing, in the preprocessing stage.

For preprocessing large quantity of data, there are usually few necessary step to take. Based on their area of competence, necessary platform and tools should be identified for the propose of processing the data. Raw data could be made available, and integrated to relational databases, as their approach. To import the raw data to database, removing invalid data (Data Cleaning) should be performed. In this stage, any inconsistencies with datatypes at a minimum will be considered.

The next stage of data cleaning usually takes place after the data is imported. This includes, verifiable inconsistencies, and missing relation between tables. Depending on the type of business, different sets of data might be process, and unrelated data can be removed from datasets to facilitate further processing. Unrelated data, might be complete sets of data, or subsets of data that the business would not have any interest in, or it is not the priority of the business.

Lastly, large quantities of data made available in the database, should be indexed for faster processing. Indexing is done based on how the database will be cleaned (e.g. unrelated data, or inconsistent data) or reports and queries generated.

For the part of this project we have imported raw data of Yelp to a MySQL database, and will do cleanups while measuring performance of the pre-processing cleanup, and conclude how this can be improved. Once the required data is ready to be processed, we will analyse data to further understand the available data and can understand different types of business in different cities and understand relation between businesses, their review and their ratings.

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

1. Introduction

As a part of assignment 1, a Yelp MySQL database was created with a Database Schema that satisfies the requests, for fulfilling the requested tasks in that assignment. For the purpose of fulfilling the requests in that assignment(assignment 1), it was sufficient to import limited data to 3 tables, including Business, Users and Review. All the raw data was imported successfully, and the details were as below:

Table	Rows	Comments	
Business	144072	# of businesses	
Users	1029423		
Review	4153150		

1. Importing more data

Target of the data analysis on in our research in narrowed to hours of operation, and categories of business. Thus, in order to address a other queries, and be able to predict other information for the purpose of this project other data have been imported to the database., as seen below.

Table	Rows	Comments
Categories	1191	# of categories
BusinessHours	653770	Hours of operation per weekday for each business per row.
Review	4153150	
BusinessCategories	0	Note: There will be further process required to fill this table
TempCategories	527604	Temporary table created by importing business id's and they category string. This will be further processed to have the business id, associate with category id.

In order to complete our dataset, executing the below query will fill the BusinessCategories table based on the values of TempCategories, and the Category table.

```
insert into businesscategories(id, business_id)
select c.id, b.business_id from tempcategory b left join categories c on b.category =
c.category
```

Now BusinessCategories table is populated, and we can drop tempCategories table;

Table	Rows	Comments
BusinessCategories	527604	

The above data should be cleaned for invalid or inconsistent data. Once verifiable data is acceptable, we will index data for faster processing.

2. Preprocessing Data

Prior to processing data, we would need to clean of the data, and index the data base for getting meaningful result, in reasonable time.

2.1. Data Cleaning & Sanity Checking

We will look into a few factors to clean data, and remove invalid and inconsistent data. Some of which are been request in the project requirement, explicitly. Later, we will provide methods to extract a sampling subset.

2.1.1. Elite Year

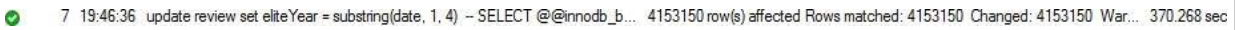
In order to have valid, and inconsistent Elite value in the database, we will remove any data prior to Yelp's establishment (2004), any future years (below query shows all users elite's year are clean from this perspective).
Also, if Elite years for any users, is prior to their first review, we will modify the Elite year to represent that year. In order to be able to query inconsistent data, we have altered the review table, add a eliteYear column wish contains the year of the review, and the data type is INT. This column will be used to accelerate lookups for consistently between, the Elite table, and Review.

select count (*) from elite where elite_year < 2004 or elite_year > 2017;

➔ 0 rows

Alter table review add column eliteYear int;
467 seconds

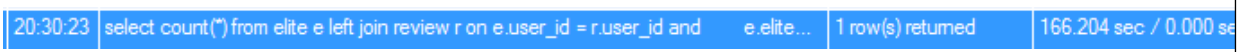
Update review set eliteYear = substring (date, 1, 4);
370 seconds



✓ Number of row in Elite Table per user, not having supporting review:

Select count (*) from elite e left join review r on e.user_id = r.user_id and e.elite_year = r.eliteYear where r.eliteYear is null;

➔ 94470 Rows (166 seconds)



➔ Remove the Elite years without supporting reivews (without Indexed tables)

SAVEPOINT withoutindex;

```
Delete elite from elite inner join(
 select * from elite e left join review r on e.user_id = r.user_id and
    e.elite_year = r.eliteYear where r.eliteYear is null;
) as X on elite.user_id = X.user_id and elite.elite_year = X.elite_year;

ROLLBACK to SAVEPOINT withoutindex;
```

✖	106	21:53:36	Delete elite from elite inner join(select e.user_id, e.elite_year from elite e left jo...	Error Code: 1317. Query execution was inter...	1905.131 sec

Remove the Elite years without supporting reviews (without Indexed tables) did not complete within 30 minutes(the query timeout was increased)

After indexing the elite, and review table the above query deleted 94470 row in **4 seconds.**

✔	110	22:42:06	ALTER TABLE elite ADD INDEX idx_elite (user_id, elite_year) USING BTREE	0 row(s) affected Records: 0 Duplicates: 0 ...	2.917 sec
⚠	111	22:42:09	ALTER TABLE elite ADD INDEX idx_review_elite_year (user_id, elite_year) US...	0 row(s) affected, 1 warning(s): 1831 Duplic...	2.964 sec
✔	112	22:43:41	show open tables where in_use>0	0 row(s) returned	0.000 sec / 0.000 sec
✖	113	22:43:41	ALTER TABLE elite ADD INDEX idx_elite (user_id, elite_year) USING BTREE	Error Code: 1061. Duplicate key name 'idx_...	0.000 sec
✔	114	22:43:55	show open tables where in_use>0	0 row(s) returned	0.000 sec / 0.000 sec
✔	115	22:43:55	ALTER TABLE review ADD INDEX idx_review_elite_year (user_id, eliteYear) U...	0 row(s) affected Records: 0 Duplicates: 0 ...	101.136 sec
✔	116	22:46:35	use yelp1	0 row(s) affected	0.000 sec
✔	117	22:46:35	Delete elite from elite inner join(select e.user_id, e.elite_year from elite e left joi...	94470 row(s) affected	3.947 sec

As seen above, 94470 Elite year did not have supporting reviews for the user, and they were remove from the Elite table withint 4 seconds with indexed tables.

2.1.2. Number of Review Per User

The number of reviews shall be consistent with the number of reviews available in the review table. For cleaning data, we will record the number of inconsistencies, and correct them. There might be some review dropped from the review table, yet the number of review count cannot be less than the number of available reviews for the users.
Below query shows there are **1191** users with review counts of less than their number of reviews in the review table.

```
select count(*) from user join (
  select r.user_id, count(user_id) r_count from review r group by user_id) as X on user.user_id
= X.user_id where user.review_count < X.r_count;
```

➔ 1191 rows

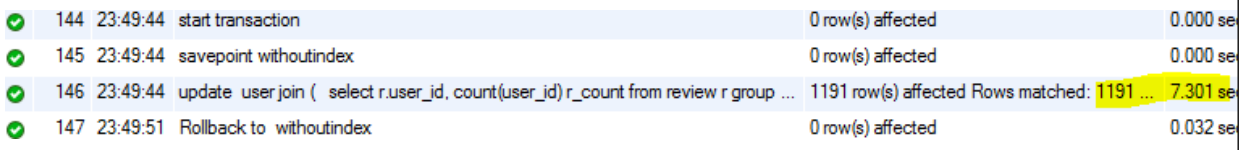
✔	133	23:32:49	select count(*) from user join (select r.user_id, count(user_id) r_count from re...	1 row(s) returned	6.178 sec / 0

✓ Update review_count for users less than the review numbers.

```
ALTER TABLE user ADD INDEX idx_user_user_id (user_id) USING BTREE;
```

UPDATE user join (
SELECT r.user_id, count(user_id) r_count from review r group by user_id) as X on user.user_id = X.user_id set review_count = X.r_count where user.review_count < X.r_count ;

➔ 1191 Rows updated in 7 seconds.



✓ Update review_count for users less than the review numbers. (Indexed tables)

START TRANSACTION;
SAVEPOINT beforePoint;
UPDATE user join (
SELECT r.user_id, count(user_id) r_count from review r group by user_id) as X on user.user_id = X.user_id set review_count = X.r_count where user.review_count < X.r_count ;
ROLLBACK TO beforePoint;

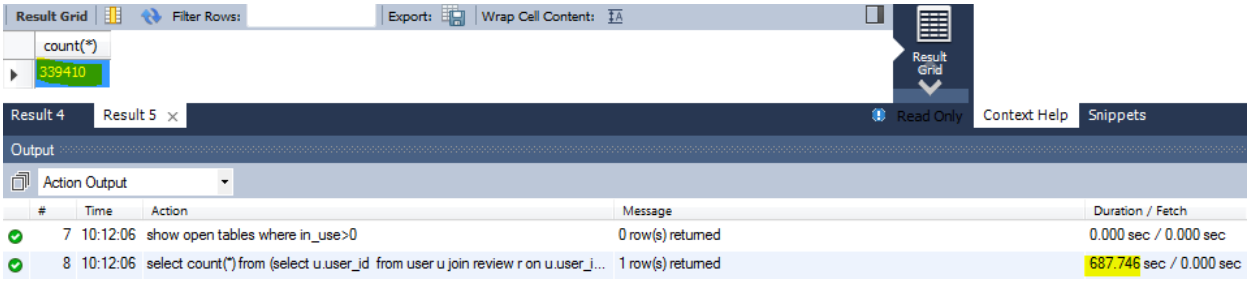
➔ 1191 Rows updated in 7 seconds.

2.2. Is this Sample Representative?

2.2.1. User average_stars

The sample data provided by Yelp, is not consistent in many ways. Since the user ratings is a crucial data for this project we will examine the average_stars for the sample users, and map the distribution on the star rating of the sample reviews.
As seen below more than 330k user or about %33 of the users have the difference of more than 0.5 points in their rating, when compared to the sample data of reviews.

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca



ALTER TABLE User add column avg_stars_in_sample float;

✔ Extract the real average stars for the sample data for comparison

UPDATE User u join (select review.user_id, avg(stars) avg_stars from review group by review.user_id) as X on u.user_id = X.user_id set u.avg_stars_in_sample = X.avg_stars ;

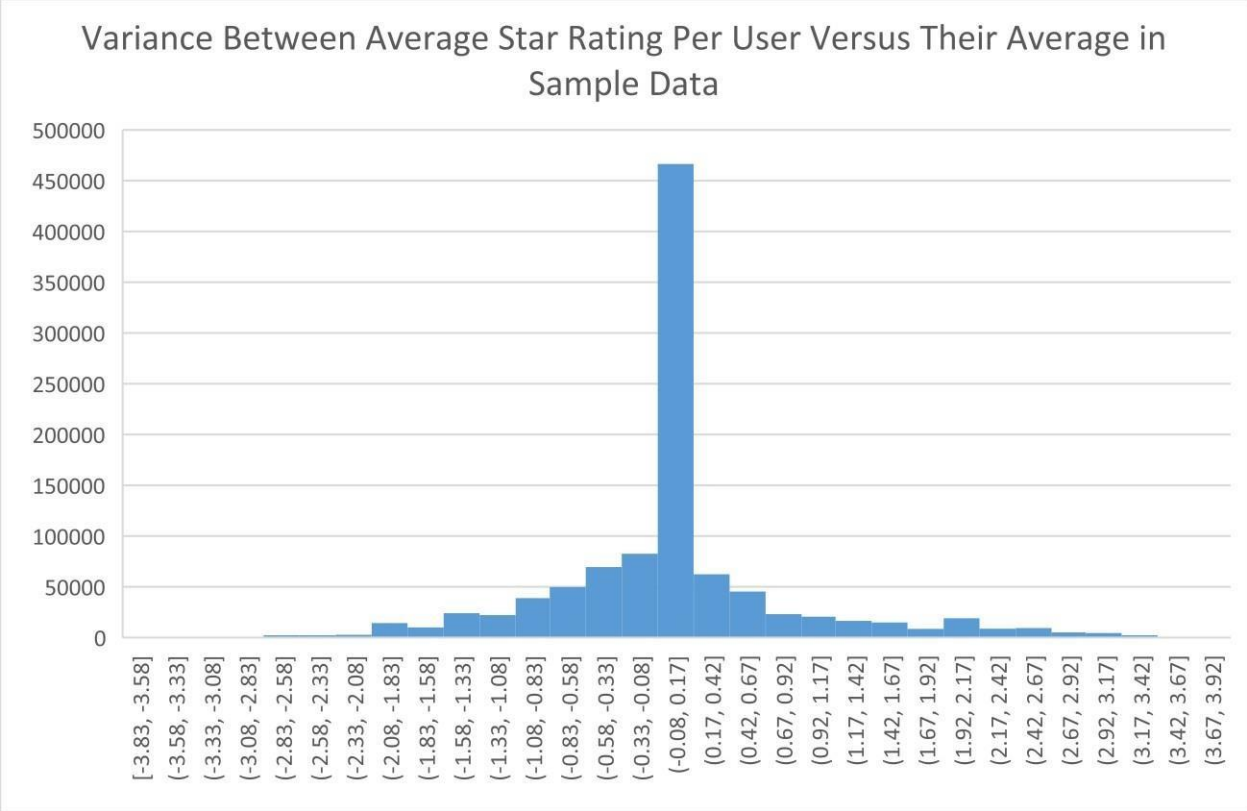
✔ Round to Only 1 decimal point

UPDATE User set avg_stars_in_sample= CONVERT(avg_stars_in_sample, DECIMAL(2,1));

✔ Export to csv file

```
SELECT user_id, average_stars, avg_stars_in_sample INTO OUTFILE
'C://ProgramData//MySQL//MySQL Server 5.7//Uploads//user-stars.csv'
   FIELDS TERMINATED BY ','
   LINES TERMINATED BY '\n'
FROM user
```

The export data was processed, and the chart below shows, the variance of actual average rating in the sample vs. what was in the user's table, using 0.25 size bins. As seen below, the data for some users might be off, and slight, yet the variance is seen in both positive, and the negative side evenly.
Thus, the sample data for average rating is considered representative.



2.3. Indexing data

2.3.1. Index for Data Cleaning

As seen in the previous section without indexing, cleaning data was almost impossible, and without reach for a relatively small table of elite. For this reason, for cleaning data, and also processing data the below indexes were make, and a brief description on the reason.

Indexing added for removing inconsistent Elite Year for users.

```
ALTER TABLE elite ADD INDEX idx_elite (user_id, elite_year) USING BTREE;
ALTER TABLE review ADD INDEX idx_review_elite_year (user_id, eliteYear) USING BTREE;
```

Indexes added for updating review_count in User table, when smaller than the number of reviews.

```
ALTER TABLE user ADD INDEX idx_user_user_id (user_id) USING BTREE;
-- review table is indexed based on user_id, above
```

Including Index for the purpose of queries based on review length.

✓ review_len field is added for study 4

```
ALTER TABLE review ADD INDEX idx_review_review_len (review_len) USING BTREE;
```

2.3.2. Index for Studying data

In order to study the widely applicable business categories, we will add for the following index to BusinessCategories table.

```
ALTER TABLE businesscategories ADD INDEX idx_businesscategories_id (id) USING BTREE;
```

For study 2 we would need to query review based on the business id, and also get their average rating for each year.

```
ALTER TABLE review ADD INDEX idx_review_years_business_id (year, business_id) USING BTREE;
```

3. Processing Data (Study Data)

3.1. Study 1 (If a business is declining or improving it's rating)

In the project, it was asked if businesses with large number of review were improving or declining in ratings. Since studying every type of business, and all the years would be out of the scope of this project; we will only target businesses, that fall in the top(widely used) categories, and narrower geographic area.

1. Top 10 Categories

```
select category, count(business_id) cat_count from categories c join businesscategories b on c.id = b.id group by category order by cat_count desc limit 10;
```

0.2 seconds (indexed data)

Category	# of businesses
Restaurants	48485
Shopping	22466
Food	21189
Beauty & Spas	13711
Home Services	11241
Nightlife	10524
Health & Medical	10476
Bars	9087

Automotive	8554
Local Services	8133

For the purpose of this Study we will target businesses in the "Restaurants" category.

2. Smaller geographic area. We will target the city with the most number of Restaurant businesses.

```
Select count(b.business_id) b_count , city from business b join businesscategories c on
b.business_id = c.business_id where c.id = 919 group by city order by b_count desc
```

Cities with the most number of Restaurant businesses.

6348 Toronto
5433 Las Vegas
3353 Phoenix
2957 Montréal
2201 Charlotte
1990 Pittsburgh
1412 Edinburgh
1356 Scottsdale
1235 Cleveland
1128 Mississauga

We will **study Restaurant business in Toronto**.

3. Target Businesses has at least 10 reviews per study year(e.g. minimum 10 reviews for 2015,and 2016, when we study decline of improvement in 2016.

We will create a new Table of BusinessRating. The data in this table populated for the samples of study.

```
CREATE TABLE BusinessRating(

    business_id VARCHAR(50),
    year2015 decimal(2, 1),
    year2016 decimal(2, 1),

    FOREIGN KEY (business_id)
REFERENCES Business(business_id));
```

✓ Insert Toronto Restaurant businesses to the temporary table

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

Insert into businessrating(business_id) select b.business_id from business b join businesscategories c on b.business_id = c.business_id where c.id = 919 and city = "Toronto";

✓ Update The above businesses with their 2015 and 2016 average rating, if they have more than 10 reviews.

Update businessrating join (select r.business_id, avg(r.stars) avg_star from review r where eliteYear = 2015 group by r.business_id having count(r.stars) > 10) as X on businessrating.business_id = X.business_id set businessrating.year2015 = X.avg_star;

Update businessrating join (select r.business_id, avg(r.stars) avg_star from review r where eliteYear = 2016 group by r.business_id having count(r.stars) > 10) as X on businessrating.business_id = X.business_id set businessrating.year2016 = X.avg_star;

select count(*) from businessrating where year2015 is not NULL and year2016 is not NULL;
1151 rows
;

1151 Toronto Restaurants have more than 10 reviews in both 2016 and 2015. These business, are the target of our study.

Businesses with Improved/Declined rating (2016 vs 2015)

select name, (year2016 - year2015) star_change from business b right join businessrating r on b.business_id = r.business_id where year2015 is not NULL and year2016 is not NULL order by star_change desc

MOST IMPROVED

TORONTO RESTAURANT	AVG RATING 2016 - 2015
Banjara Indian Cuisine	1.4
Horizons Restaurant	1.1
Dino's Wood Burning Pizza	1.1
Arisu	1.1
Joe Badali's Ristorante Italiano & Bar	0.9
Pho Nam Dinh	0.9
Foxley Bistro and Bar	0.9
Fune Japanese Restaurant	0.9
Marben	0.9

Nirvana	0.9

select name, (year2016 - year2015) star_change from business b right join businessrating r on b.business_id = r.business_id where year2015 is not NULL and year2016 is not NULL order by star_change

MOST DECLINED
TORONTO RESTAURANT AVG RATING 2016 - 2015

Gyutaro	-1.8
King Palace	-1.5
The Miller Tavern	-1.4
Qi Sushi	-1.3
309 Dhaba Indian Excellence	-1.3
Fat Bastard Burrito	-1.3
Blu Ristorante	-1.3
Chacho's Fine Mexican Dining	-1.2
Madison Avenue Pub & Restaurant	-1.2

The above study shows in the list of Toronto's Restaurant with more than 10 rating in 2015, and 2016, *Gyutaro* has the most decline, and *Banjara Indian Cuisine* has the most improved rating, 2016 vs the previous year.

3.4. Study 2 (Predict what rating a user will give based on his previous ratings)

We would have to understand the rating pattern or distribution for other users, and be able classify all users for this study.

For this purpose, we will study randomly, 1 user's rating pattern in 2015 by comparing his ratings to the average rating of a widely-rated restaurant, and based on that predict the user's rating in 2016, and compare the results, and we will expand our study to others. The target user will have about 50 reviews a year (or about 1 per week) for 2015. The following user is chosen randomly.

```
Select count(*) from review where user_id = 'DeVGAiOf2mHVUDfxvuhVlQ' and eliteYear = 2016;
```

User_id	name	2015 reviews	2016 reviews
DeVGAiOf2mHVUDfxvuhVlQ	Jay	54	32

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

The outcome of the study to include three(3) rating quality per user. Each user will have three statistic based on how they rank each category(percentage of accuracy), and Yelp would determine if it should show a particular users' rating and review for a particular business, based on the average start rating of the business.

The 3 quality statistics of user rating include:

- High accuracy
  - How well does the user rate restaurant with an average of 3.5 star rating or more
- Mid accuracy
  - How well does the user rate restaurant with an average of 2.5 to 3.5 star rating or more
- Low accuracy
  - How well does the user rate restaurant with an average of 2.5 star rating or less

As below, we will create a temporary table, and study the above user for 2015, and compare the pattern with his ratings in 2016, to conclude this strategy for classifying the quality of user's ratings.

✓ Create a temporary table to hold the businesses the users rated in one year.

```
CREATE TABLE tempBusiness(business_id VARCHAR(50), star_year INT, avg_stars
DECIMAL(2,1), user_stars DECIMAL(2,1));
```

✓ Add Businesses that our subject user rated in 2015, and his rating.

```
Insert into tempBusiness (business_id, user_stars) select business_id, stars from review where
eliteYear = 2015 and user_id = 'DeVGAiOf2mHVUDfxvuhVlQ'
```

✓ Update rows on this table with average star rating for 2015, for this businesses.

```
Update tempBusiness join (select r.business_id, round(avg(r.stars),1) avgRating from review r
where eliteYear = 2015 group by r.business_id ) as X on X.business_id =
tempBusiness.business_id set tempBusiness.avg_stars = X.avgRating;
```

✓ Rate the quality of User's rating based on high, mid, low average ratings of restaurant. After some experiments the below, it was clear some overlap on the user's rating would give us a clearer picture.

```
Select round((select count(*) from tempbusiness where avg_stars > 3.5 and user_stars>=4)/
      (select count(*) from tempbusiness where avg_stars > 3.5), 2) as
stars_high_user_accuracy,
```

```
  round((select count(*) from tempbusiness where avg_stars > 2.5 and avg_stars <= 3.5 and
user_stars <=4 and user_stars >=2)/
      (select count(*) from tempbusiness where avg_stars > 2.5 and avg_stars <= 3.5), 2) as
stars_mid_user_accuracy,
  round((select count(*) from tempbusiness where avg_stars <= 2.5 and user_stars <= 3)/
      (select count(*) from tempbusiness where avg_stars <= 2.5 ), 2) as
stars_low_user_accuracy;
```

✓ Below show's our subject user's review in 2015 is about %70 worthy for businesses having average of 3.5 stars of more. %68 worthy for businesses having average of 2.5 to 3.5 stars. And %20 worthy for businesses having average of 2.5 stars or less. Base on result from other users, if the quality of this user in any of the 3 classes fall above the threshold it can be consider worthy.

stars_high_user_accuracy	stars_mid_user_accuracy	stars_low_user_accuracy
0.70	0.68	0.20

Now we will compare the quality of our subject user's review with his 2016 ratings, by applying the same logic.

```
drop table tempBusiness;

CREATE TABLE tempBusiness(business_id VARCHAR(50), star_year INT, avg_stars
DECIMAL(2,1), user_stars DECIMAL(2,1));

insert into tempBusiness (business_id, user_stars) select business_id, stars from review where
eliteYear = 2016 and user_id = 'DeVGAiOf2mHVUDfxvuhVlQ';

update tempBusiness join (select r.business_id, round(avg(r.stars),1) avgRating from review r
where eliteYear = 2016 group by r.business_id ) as X on X.business_id =
tempBusiness.business_id set tempBusiness.avg_stars = X.avgRating;
```

✓ Below show's our subject user's review in 2016 is about %93 worthy for businesses having average of 3.5 stars of more. %67 worthy for businesses having average of 2.5 to 3.5 stars. And %100 worthy for businesses having average of 2.5 stars or less. Base on result from other users, if the quality of this user in any of the 3 classes fall above the threshold it can be consider worthy.

stars_high_user_accuracy	stars_mid_user_accuracy	stars_low_user_accuracy
0.93	0.67	1.00

Review Quality Category	2015 star ratings	2016 star rating
Business with 3.5 or more average stars	%70	%93
Business with 2.5 to 3.5 average stars	%68	%67
Business with 2.5 or less average stars	%20	%100

The above user show to provide good reviews and rating for business in category 1 and 2.
He can build up on category 3 reviews, and improve inconsistencies seen in his 2015 rating, over a period of few years.
Our quality rating recommends having this user's review for categories 3 towards the end of the list of reviews.

Using the above logic, every user can be classified per year on how solid, and close their rating is to the average reviews, and where should their review be show on Yelp website.

3.4. Study 3 (Understanding if business hours reflect the star rating)

For this study, we will classify businesses based on their hours of operation, and study if the rating of any group is higher or lower than the average rating of all businesses. We will only businesses in the **Restaurants** category as different categories might impact the star ratings, as well.

```
Select count(X.business_id) from (select distinct(business_id) from businesshours) as X join businesscategories c on X.business_id = c.business_id where c.id = 919;
```
➔ 34268 rows (Number of Restaurants with hours of operation)

```
Select avg(b.stars) from (select X.business_id from (select distinct(business_id) from businesshours ) as X join businesscategories c on X.business_id = c.business_id where c.id = 919) as Y join business b on Y.business_id = b .business_id ;
```
➔ 3.57 (Average Star Rating for all the Restaurants with hours of operation)

- Query Restaurants closing after midnight are as below:

```
Select count(X.business_id) from (select distinct(business_id) from businesshours where hour_close < 5) as X join businesscategories c on X.business_id = c.business_id where c.id = 919
```
➔ 9689(Number of Restaurants closing on or after midnight)

```
Select avg(b.stars) from (
select X.business_id from (select distinct(business_id) from businesshours where hour_close < 5) as X join businesscategories c on X.business_id = c.business_id where c.id = 919
) as Y join business b on Y.business_id = b .business_id ;
```

➔ 3.35 (Average Star Rating for all the Restaurants closing on or after midnight)

- Query Restaurants serving early breakfast

```
Select count(X.business_id) from (select distinct(business_id) from businesshours where hour_open < 8) as X join businesscategories c on X.business_id = c.business_id where c.id = 919
```

➔ 5931 Number of Restaurants serving early breakfast

```
Select avg(b.stars) from (
select X.business_id from (select distinct(business_id) from businesshours where hour_open < 8) as X join businesscategories c on X.business_id = c.business_id where c.id = 919
) as Y join business b on Y.business_id = b .business_id ;
```

➔ 3.47 (Average Star Rating for all the Restaurants serving early breakfast)

The number of businesses having hours of operation in the database is limited to 34268. We will only study these businesses.

		Count	Average Stars
Overall	All Restaurants	34268	3.57
Category 1	Restaurants closing on after midnight (at least 1 day in week)	9686	3.35
Category 2	Restaurants opening on or before 8 am(Serving Breakfast)	5931	3.47

The above shows, business opening early and closing later after midnight generally having a lower star rating than average business. Meaning business opening later, and closing before midnight have generally higher ratings.

3.4. Study 4 (Would review length affect how other perceive the review)

In this study will analyse how review length affect the rating of the review itself. First we will include the review length, in the review table.

```
ALTER TABLE review ADD COLUMN review_len INT;

UPDATE review set review_len = CHAR_LENGTH(text);
```

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

Now we break now reviews to buckets based on the review length, and query average usefulness, coolness, and being funny for each bucket.

```
SELECT avg(useful), avg(funny), avg(cool) from review where review_len < 500;
 -> 0.5259 0.1884 0.2563

SELECT avg(useful), avg(funny), avg(cool) from review where review_len >= 500 and
review_len < 1000;
 -> 1.1653 0.4704 0.6061

SELECT avg(useful), avg(funny), avg(cool) from review where review_len >= 1000 and
review_len < 2000;
 -> 2.0076 0.9230 1.1131

SELECT avg(useful), avg(funny), avg(cool) from review where review_len >= 2000 and
review_len < 4000;
 -> 3.1745 1.5472 1.7066


SELECT avg(useful), avg(funny), avg(cool) from review where review_len >=
 -> 4.6218 2.2838 2.3138
```

Review length bucket (characters)	Avg. Useful	Avg. Cool	Avg. Funny	
< 500	0.5259	0.1884	0.2563	
>= 500 && < 1000	1.1653	0.4704	0.6061	
>= 1000 && < 2000	2.0076	0.9230	1.1131	
>= 2000 && < 4000	3.1745	1.5472	1.7066	
>= 4000	4.6218	2.2838	2.3138	

The above chart is the clear indication that longer reviews are perceived better, and the average number of usefulness, coolness and being funny trends upward.

4. Further Analysis of the dataset

The below study is completed on a subset of database.

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

As the variety of people choose their favourite restaurant, business, airport, ticket.etc through the yelp website. Analyze the database of yelp website give the important information to customers. In the business table receive ten or thousands of reviews. However, there are lots of reviews, readers prefer to look at the star rating and ignore the review text. For example, a customer give three stars to a restaurant because of does not have enough capacity of parking lot and have a poor air condition. On the other hand, another customer who give three stars to the same restaurant, and he does not really consider the capacity of parking lot or poor air condition in his ratings. Some customer give three stars because the quality of food, dose not satisfy. In this example that's important to read the review text. Now, In the categories in business table to illustrate the Distribution of Business Categories through these quires.

```
select
categories_001,categories_002,categories_003,categories_004,categories_005,categories_
_006,categories_007,categories_008,categories_009,categories_010,count(*)/42.153 count
from business group by
categories_001,categories_002,categories_003,categories_004,categories_005,categories_
_006,categories_007,categories_008,categories_009,categories_010 order by count desc;
```

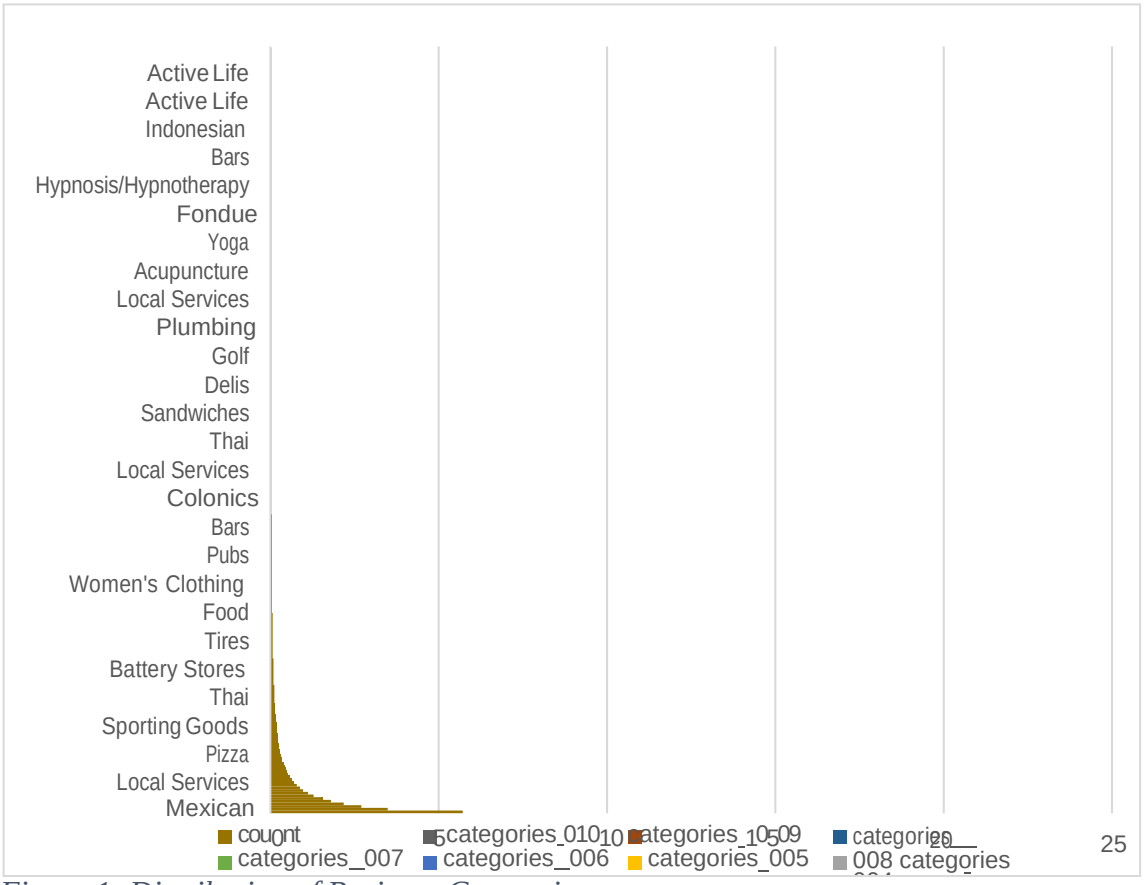


Figure 1: Distribution of Business Categories

Referring to figure1, it shows that the distribution of restaurant in business table is much higher than the distribution of other facilities.

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

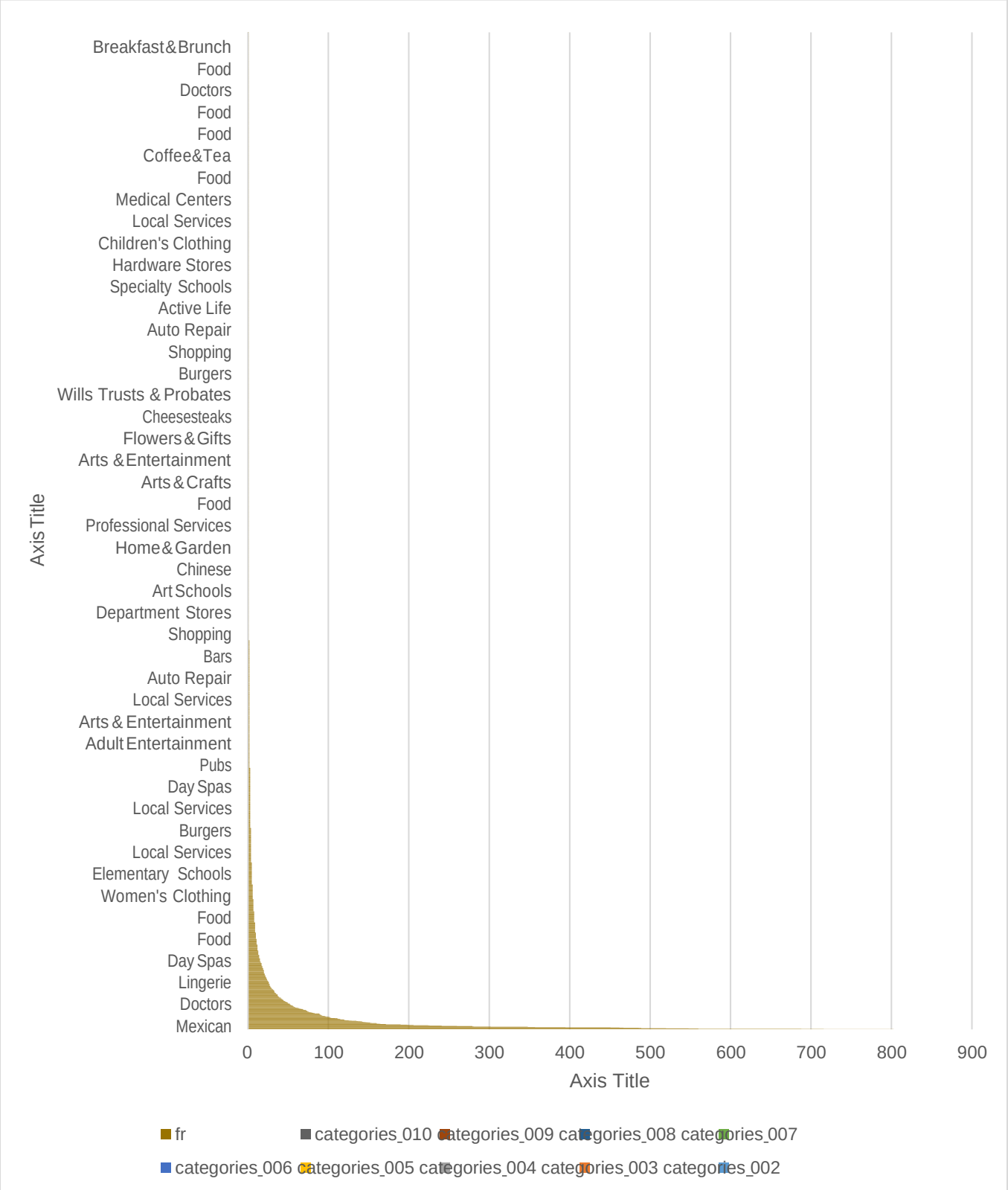


Figure 2: Distribution of Reviews

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca

Since restaurants have the highest range of distribution in the business table, they were received high range of reviews by customers as well. Figure2 illustrate the distribution of reviews which is related to restaurants in review table.

```
Select
categories_001,categories_002,categories_003,categories_004,categories_005,categories_
_006,categories_007,categories_008,categories_009,categories_010,count(review_count) fr
from business group by
categories_001,categories_002,categories_003,categories_004,categories_005,categories_
_006,categories__007,categories__008,categories__009,categories__010 order by fr desc;
```

Figure1, shows the distribution of business categories in the dataset. Restaurants make up almost 8.03% of the 42.153 businesses. Also, 803 of 1.125.458 text reviews are about restaurants as shown by figure2. Figure3 shows that the star rating (out of 5) for the restaurant reviews are not consistently distributed.

```
select distinct count1+count2+count3+count4+count5+count6+count7+count8+count9+count10
,q1.stars from(select count(*)/100 count1,stars from business where
categories_002='Restaurants' group by stars)q1 join
(select count(*)/100 count2,stars from business where categories_002='Restaurants' group by
stars)q2 join
(select count(*)/100 count3,stars from business where categories_003='Restaurants' group by
stars)q3 join
(select count(*)/100 count4,stars from business where categories_004='Restaurants' group by
stars)q4 join
(select count(*)/100 count5,stars from business where categories_005='Restaurants' group by
stars)q5 join
(select count(*)/100 count6,stars from business where categories_006='Restaurants' group by
stars)q6 join
(select count(*)/100 count7,stars from business where categories_007='Restaurants' group by
stars)q7 join
(select count(*)/100 count8,stars from business where categories_008='Restaurants' group by
stars)q8 join
(select count(*)/100 count9,stars from business where categories_009='Restaurants' group by
stars)q9 join
(select count(*)/100 count10,stars from business where categories_010='Restaurants' group by
stars)q10 group by stars;
```

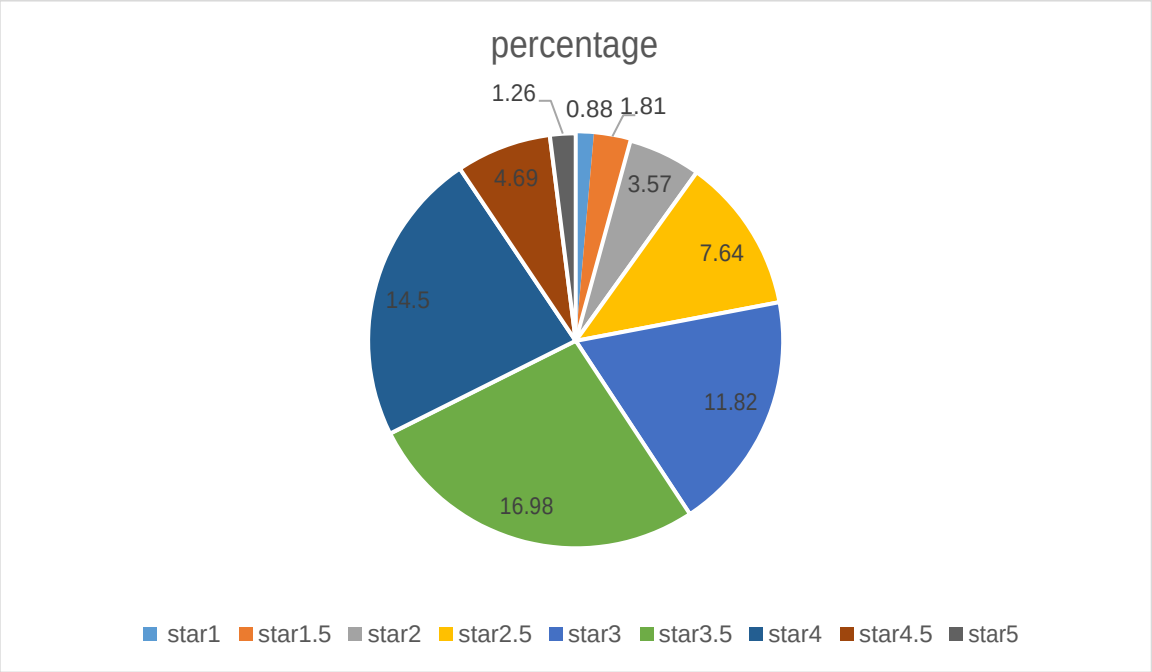


Figure 3 : Stars Distribution for Restaurant Reviews

The distribution of restaurants is higher than other facilities in business and review tables. The next step to analysis the star distribution for restaurant reviews, star 3.5 have the highest range among stars. The next highest range is related to star4. On the other hand, star1 have the lowest range among stars. As result, Figure3 shows that the positive points of restaurants, which were given by customers are weighted than negative points of them.

Refereeing to figure4, which shows that the votes_cool are given by customers have changed through the seasons and specific dates. For example, on '2016-02-15', in February is received the high votes_cool.
These statistics show that customers around the Christmas time or in the summer probably on the vacation they voted the most positive votes. Also, on '2015-02-15' in February, on ‘2014-12-29’ in December, on ‘2013-08-19’ in August, on ‘2012-01-03’ in January, on 2011-08-16 in August, on ‘2010-12-30’ in December, on ‘2009-01-05’ in January, on ‘2008-07-08’ in July, on ‘2007-07-26’in July, on ‘2006-01-13’ in January, on ‘2005-07-21’ in July, on ‘2004-10-12’ in October were received the highest votes_cool.

select count(votes_cool) count, date from review1 group by date order by count desc;

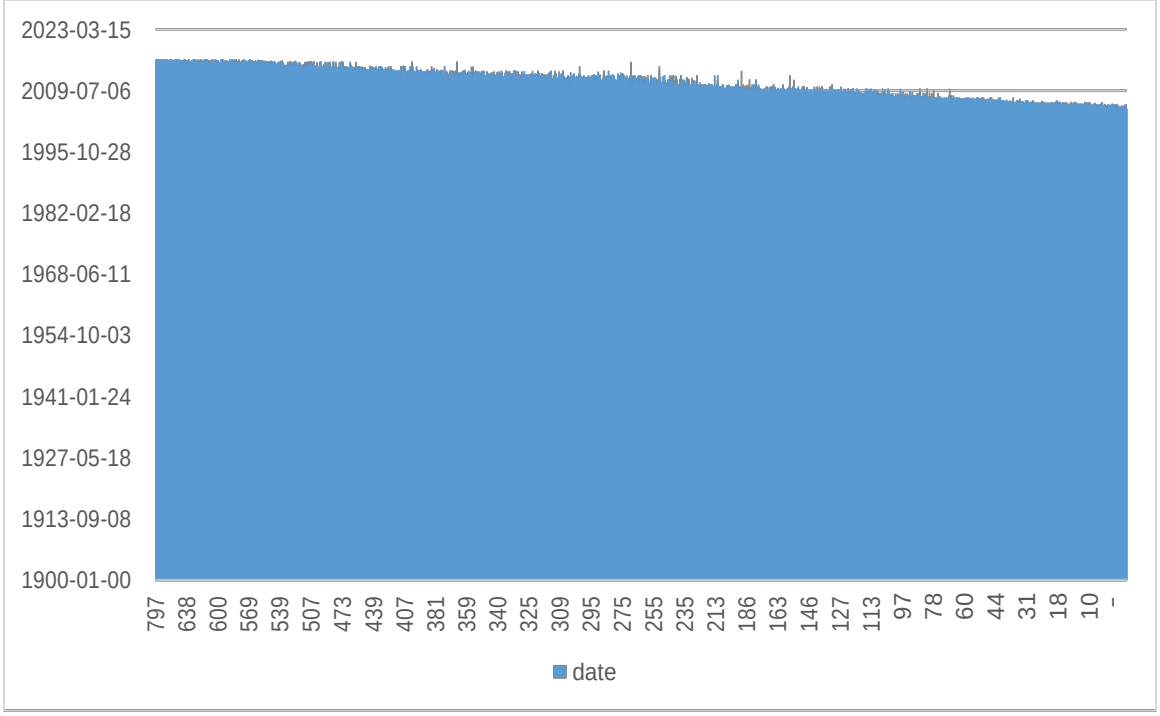


Figure 4: Distribution of count(votes_cool) through the years

Also, refereeing to figure5 which shows that the votes_useful are given by customers were changed through the seasons and specific dates. Same Like the pervious query votes_useful are given by customers have changed through the seasons and specific dates. For example, on '2016-02-15' in February is received the highest votes_useful. Also in '2015-02-15' in February, on 2014-12-29 in December around the Christmas time or in the summer customers voted the most positive votes. The below query show the distribution of votes_useful in review table.

select count(votes_useful) count, date from review1 group by date order by count desc;

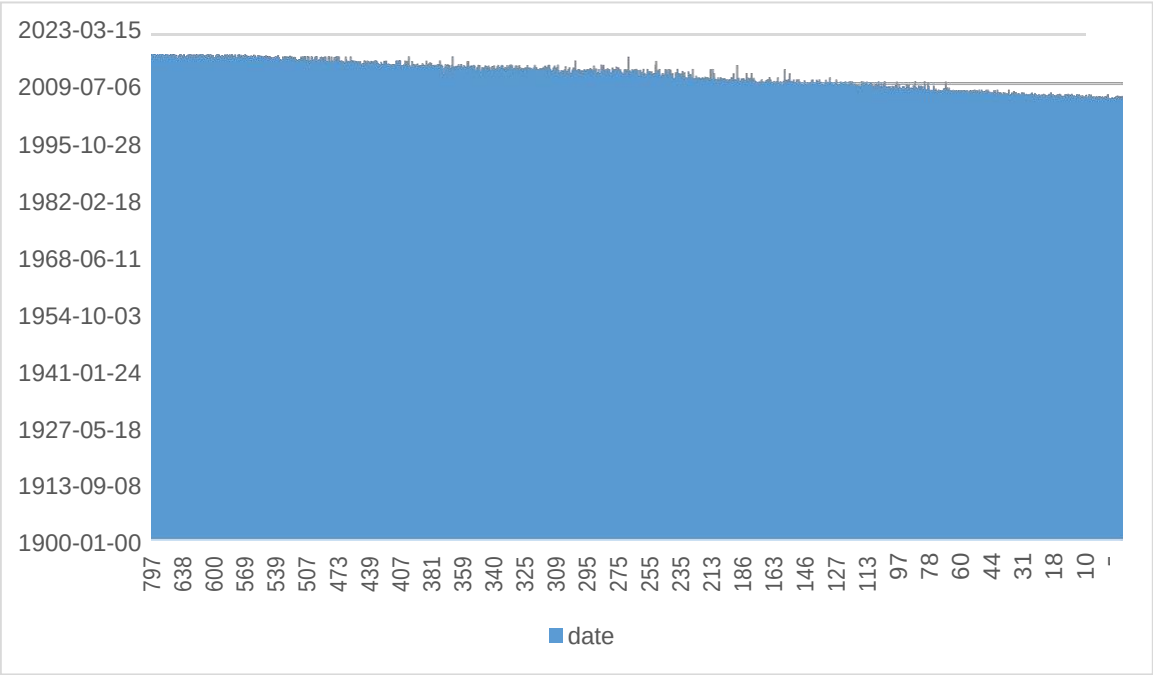


Figure 5: Distribution of count(votes_useful) through the years in review table

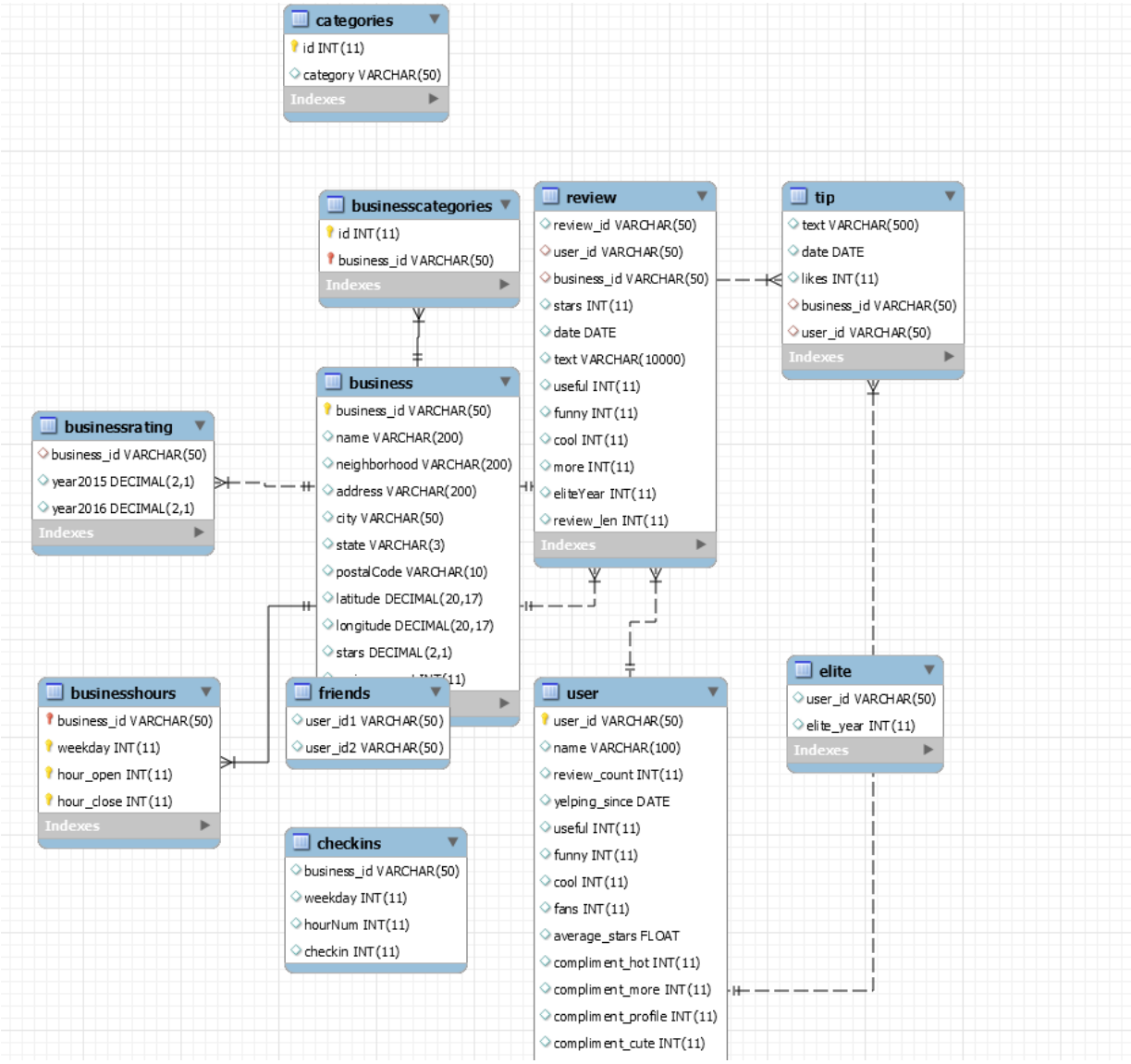
4. EER Diagram

The database for the purpose of this study was normalized. The snapshot of the database is seen below. As can be seen for many to many relation between business and categories, a business categories table has been added. Tables tip, review, business hours and business categories have foreign key relationship with business table on business_id.

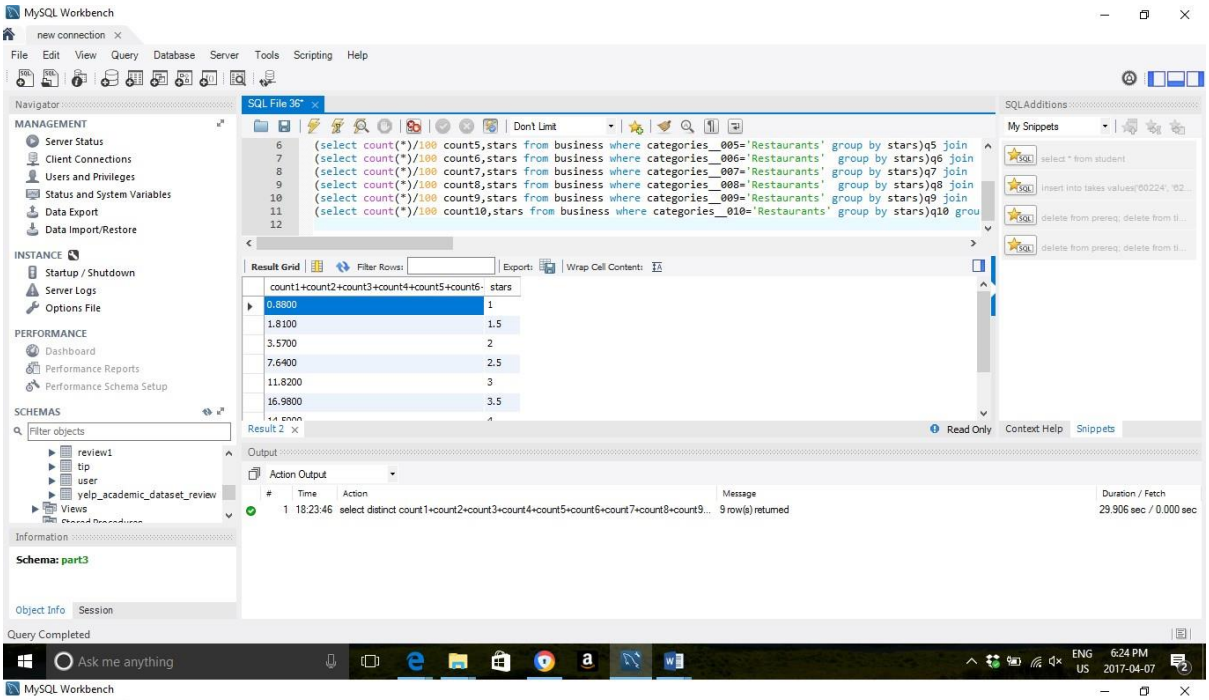
Table tip, and review friends, have foreign key relation with user on users_id.

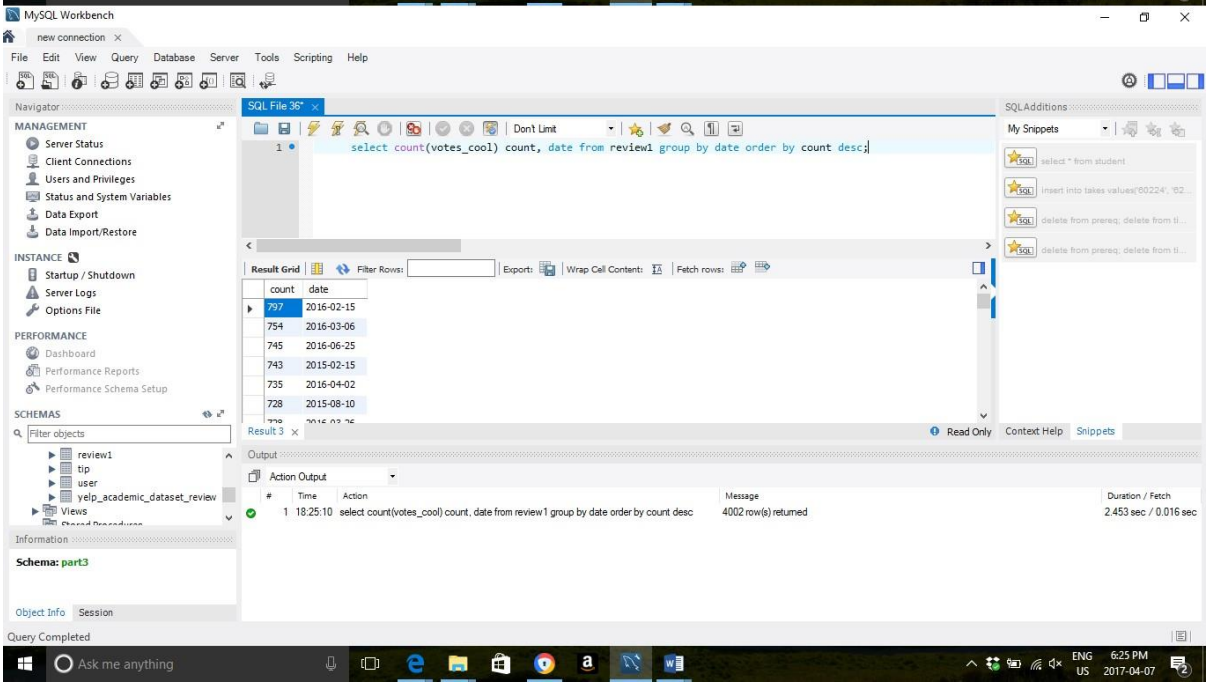
Business hour table has any number of open and closed hour per business, and as long as they are unique.

Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca



Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca





Farnoush Yousefi Kejani fyousefikejani@uwaterloo.ca