Project Report: Yelp Dataset

Group 2.

Adriana Marcela Penaranda Baron

Carlos Rey Pinto

Eduardo Roberto Williams Cascante

Haldo Jose Somoza Solis

Ishika Sukhija

Luis Alejandro Gutierrez Hayek

Marzieh Mohammadi Kokaneh

Nilesh Khurana

Lambton College

Big Data Analytics

BDM 1024 - Data Technology Solutions

To: Mr. Bhavik Gandh

Mississauga, Ontario

16 Aug, 2023

Table of Contents

Project Report: Yelp Dataset

Data Model Diagram

The dataset selected is the "Yelp Open Dataset," available as JSON files on the Yelp website and with a size of 8.65GB uncompressed. (yelp, 2023)

Entities

First, the following entities were identified in the selected Yelp company dataset:

- Businesses: Contains information related to companies, like name, city, address, etc.
- Attributes: Contains different features related to businesses.
- Categories: Contains the types of businesses
- Hours: Contains schedules, including days of the week and hours that operate the business.
- Check-ins: Enumerates check-ins for a business.
- Photos: This page contains some pictures of the businesses.
- Users: This contains the information of the customers that review or leave tips to improve the business service.
- Users Elite: This contains the number of years that the user has had elite status.
- Friends: Contains the relationships between users.
- Tips: Contains a short recommendation that users give to businesses for them to improve their services.
- Reviews: Contains full review text data, including the user that wrote the review and the business the review is written for.

The following table describes the relationships between "Businesses" entities:

Entity	Relationship type	Cardinality
Attributes	Composite association regarding Businesses	Businesses → Attributes: One to many
Categories	Composite association regarding Businesses	Businesses → Categories: One to many
Hours	Composite association regarding Businesses	Businesses → Hours: One to many
Checkins	Composite association regarding Businesses	Businesses → Checkins: One to many
Photos	Composite association regarding Businesses	Businesses → Photos: One to many

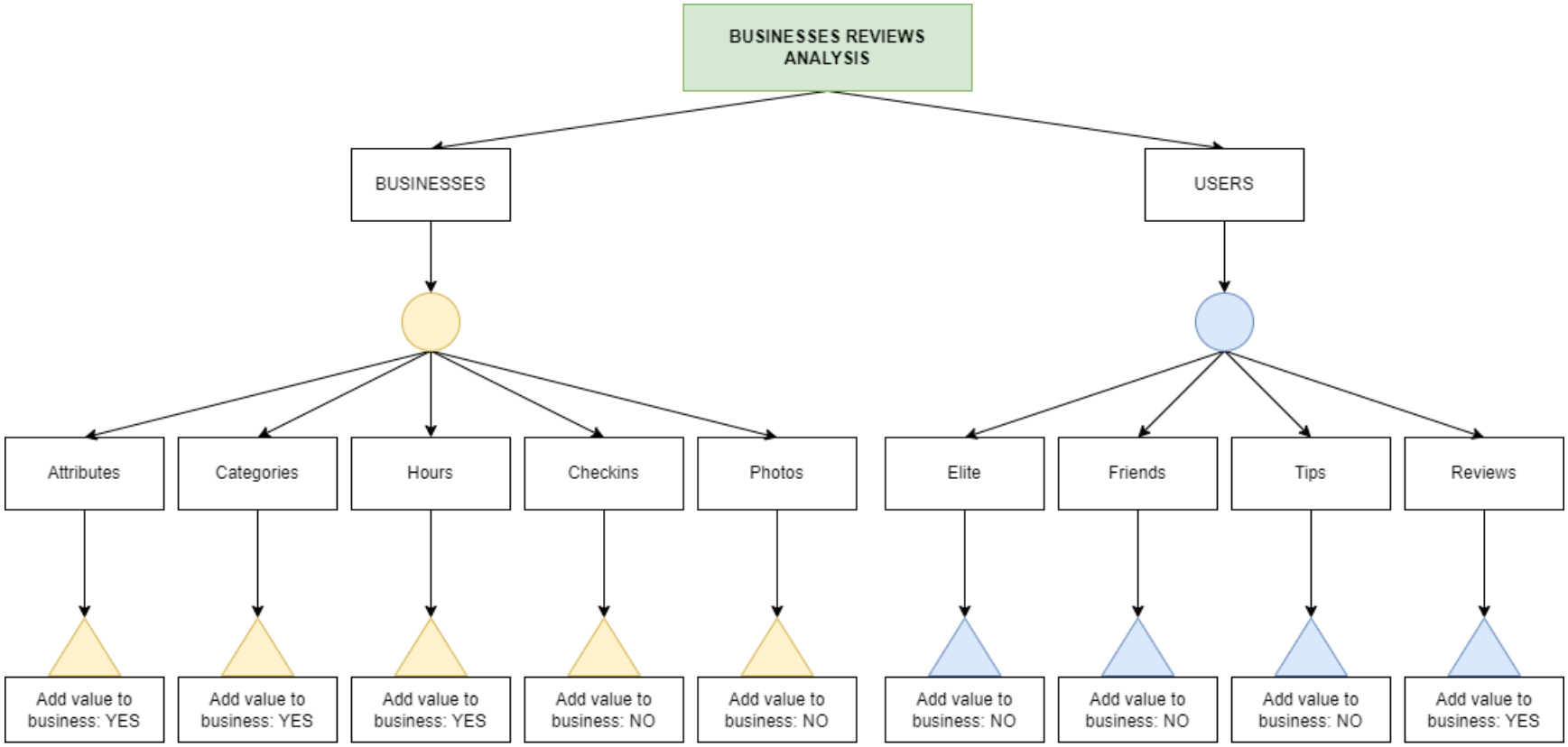
The following table describes the relationships between "Users" entities:

Entity	Relationship type	Cardinality
Users "elite"	Composite association regarding Users	Users → Users elite: One to many
Friends	Composite association regarding Users	Users → Friends: One to many

In addition, it was identified an aggregation association between "Businesses" and "Users" entities, which is many to many. This relationship is represented through Tips and Reviews entities, as follow:

Entity	Relationship type	Cardinality
Tips	Composite association regarding Users and Businesses	Users → Tips: One to many Businesses → Tips: One to many
Reviews	Composite association regarding Users and Businesses	Users → Reviews: One to many Businesses → Reviews: One to many

Decision Tree Analysis



Expected value

We looked for information about businesses, which added value to them and we consider the following:
Attributes: Adds value because users can know features about businesses and what they offer
Categories: Adds value because users can classify business according to their needs and what they are looking for
Hours: Definitely it adds value because users must know attention schedules
Checkins: We consider date and time of checkins on a business are not important to our analysis
Photos: We consider photos are important to another kind of analysis

Expected value

We looked for information about users, which added value to businesses reviews and we consider the following:
Elite and friends: We consider elite and friends entities are useful if we wanted to make some analysis about users information, so we discarded these entities.
Tips: We discarded tips entity too, because we deem we can find the same information in the Reviews table, which offers more useful and accurate data for our analysis.

Following a qualitative decision tree analysis, it was agreed to work with the following 6 entities, which contain the main features of the businesses and meet all requirements for SQL database importation:

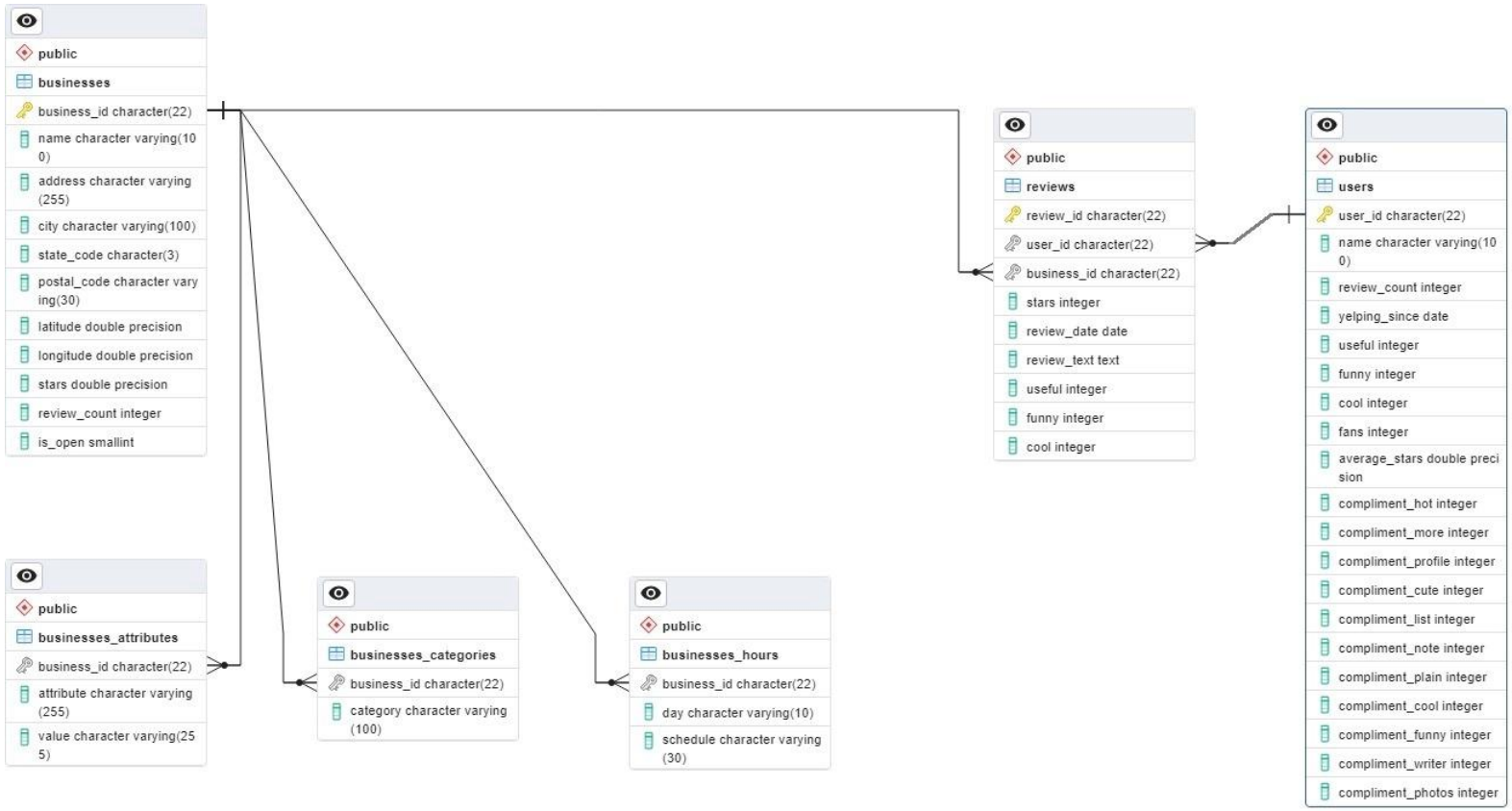
- Businesses
- Attributes
- Categories
- Hours
- Users
- Reviews

The following table contains the main reasons why the entities about business and users were or were not selected:

Entity	Reason
Businesses	The principal entity that contains the main description columns about the business
Attributes	This table adds value to the data because users can learn about the businesses features and what they offer, and it would be possible to analyze the preferred features for users and categorize them.
Categories	It is very important to know how businesses are classified according to what they offer.
Hours	It is important that users know the business hours, and the most demanded business hours can be part of the analysis.
Check-ins	This entity will be considered in the final project to process it in a NoSQL Database
Photos	Photos are very important to the Yelp database, but they do not contribute to this SQL database analysis.
Elite and friends Tips	These entities will probably be considered in the final project to process it in a NoSQL Database based on the type of information it contains.

Note: Attributes, integrity rules like PK and FK, and Indices can be found in the data dictionary.

Entity relationship diagram (ER Diagram)



Note: ER Diagram was creating using PgAdming4

Yelp Dataset Dictionary

The dataset dictionary was created. This step is crucial for the creation of the database and the safe import of the data, always maintaining the information integrity. This data dictionary was created considering the Yelp documentation provided by each entity and serves as a valuable reference since it contains details such as the variables, data types, definitions, and some examples. This documentation facilitates the analysis of the data, minimizes misinterpretation of the data since there is a better understanding of the information, and allows more reliable and effective conclusions and recommendations to be reached.

businesses					
Description:	Contains business data including location data, attributes, and categories				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
business_id	char(22)	PK	not null	business id	tnhfDv5Il8EaGSXZGiuQGg
name	varchar(100)		null	the business's name	Garaje
address	varchar(255)		null	the full address of the business	475 3rd St
city	varchar(100)		null	the city	San Francisco
state_code	char(3)		null	state code	CA
postal_code	varchar(30)		null	the postal code	94107
latitude	float		null	latitude	37.7817529521
longitude	float		null	longitude	-122.39612197
stars	float		null	star rating, rounded to half-stars	4.5
review_count	int		null	number of reviews	1198
is_open	smallint		null	0 or 1 for closed or open, respectively	1
Index	businesses_idx0 (name) businesses_idx1 (state_code, city, is_open)				

businesses_attributes					
Description:	Attributes and subattributes of businesses				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
business_id	char(22)	FK	not null references businesses (business_id)	business id	tnhfDv5Il8EaGSXZGiuQGg
attribute	varchar(255)		not null	Attribute:Subattribute description	RestaurantsTakeOut BusinessParking:garage
value	varchar(255)		not null	true / false	
Index	businesses_attributes_idx0 (business_id, checked)				

businesses_categories					
Description:	Categories of businesses				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
business_id	char(22)	FK	not null references businesses (business_id)	business id	tnhfDv5Il8EaGSXZGiuQGg
category	varchar(100)		not null	category description	Mexican Burgers
Index	businesses_categories_idx0 (business_id, category)				

businesses_hours					
Description:	Businesses schedules				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
business_id	char(22)	FK	not null references businesses (business_id)	business id	tnhfDv5Il8EaGSXZGiuQGg
day	varchar(10)		not null check(day in ('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'))	day of week	Tuesday
schedule	varchar(30)		not null	hour from - hour to	10:00-21:00
Index	businesses_hours_idx0 (business_id)				

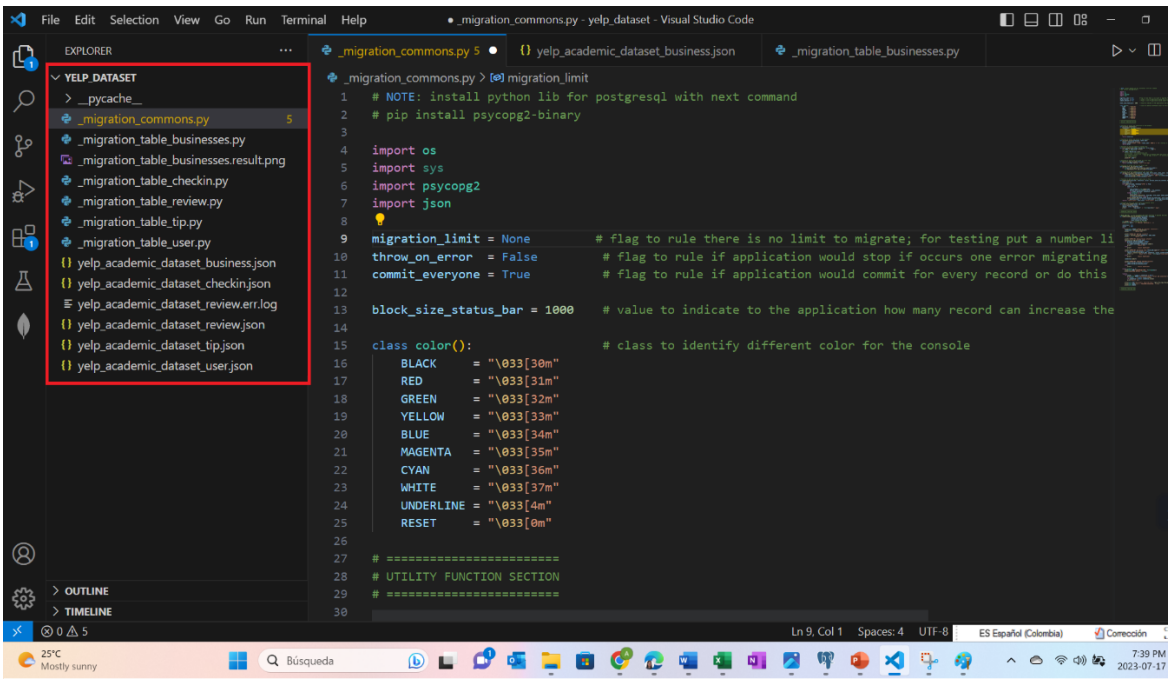
Users					
Description:	User data including all the metadata associated with the user				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
user_id	char(22)	PK	not null	user id	Ha3iJu77CxlrFm-vQRs_8g
name	varchar(100)		null	the user's first name	Sebastien
review_count	int		null	the number of reviews they've written	56
yelping_since	date		null	when the user joined Yelp, formatted like YYYY-MM-DD	2011-01-01
useful	int		null	number of useful votes sent by the user	21
funny	int		null	number of funny votes sent by the user	88
cool	int		null	number of cool votes sent by the user	15
fans	int		null	number of fans the user has	1032
average_stars	float		null	average rating of all reviews	4.31
compliment_hot	int		null	number of hot compliments received by the user	339
compliment_more	int		null	number of more compliments received by the user	668
compliment_profile	int		null	number of profile compliments received by the user	42
compliment_cute	int		null	number of cute compliments received by the user	62
compliment_list	int		null	number of list compliments received by the user	37
compliment_note	int		null	number of note compliments received by the user	356
compliment_plain	int		null	number of plain compliments received by the user	68
compliment_cool	int		null	number of cool compliments received by the user	91
compliment_funny	int		null	number of funny compliments received by the user	99
compliment_writer	int		null	number of writer compliments received by the user	95
compliment_photos	int		null	number of photo compliments received by the user	50
Index	users_idx0 (name) users_idx1 (yelping_since)				

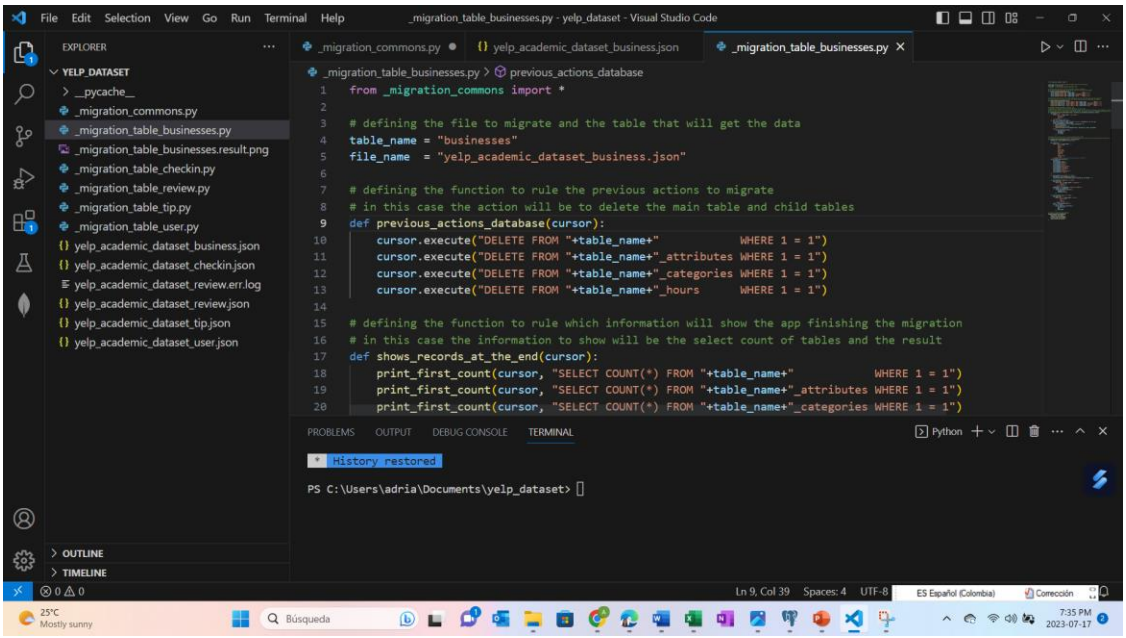
Reviews					
Description:	Contains full review text data including the user_id that wrote the review and the business_id the review is written for				
Field name	**Data type**	**Key**	**Constraint**	**Description**	**Example**
review_id	char(22)	PK	not null	review id	zdSx_SD6obEhz9VrW9uAWA
user_id	char(22)	FK	not null references users (user_id)	user id	Ha3iJu77CxlrFm-vQRs_8g
business_id	char(22)	FK	not null references businesses (business_id)	business id	tnhfDv5Il8EaGSXZGiuQGg
stars	int		null	star rating	4
review_date	date		null	date formatted YYYY-MM-DD	2016-03-09
review_text	text		null	the review itself	Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks
useful	int		null	number of useful votes received	0
funny	int		null	number of funny votes received	0
cool	int		null	number of cool votes received	0
Index	reviews_idx0 (user_id, review_date) reviews_idx1 (business_id, review_date) reviews_idx2 (review_date, business_id, user_id)				

Yelp Dataset Analysis in a PostgreSQL Database

Python program created to transform the information from json format to PostgreSQL

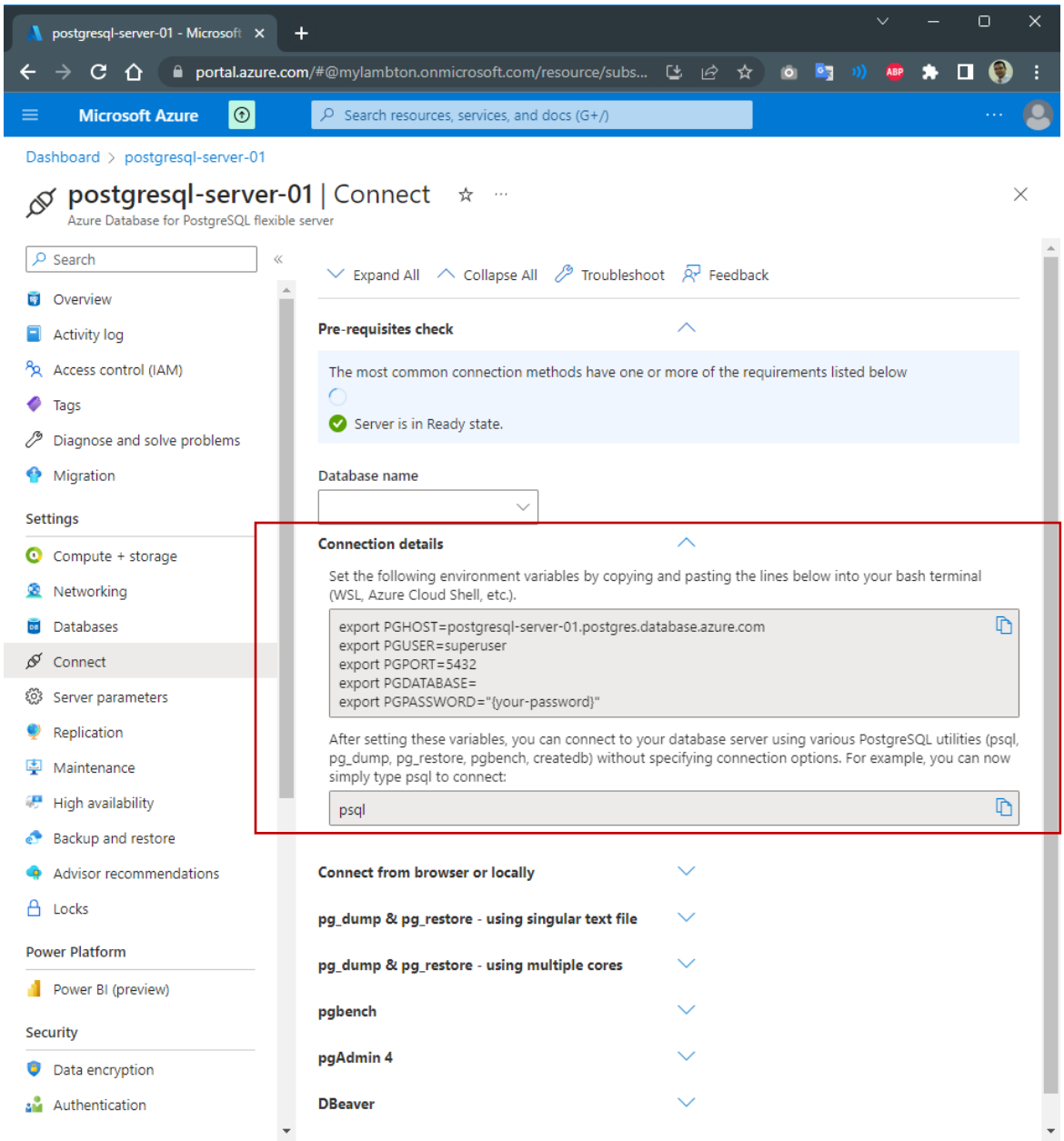
It was necessary to create 7 python programs to transform the dataset from json format to PostgreSQL. The file "_migration_commons.py" contains the main features of the data transformation and connection. Also, it was created one py file for each entity, as shown below:





PostgreSQL database and tables with relationships created in the Cloud (Azure).

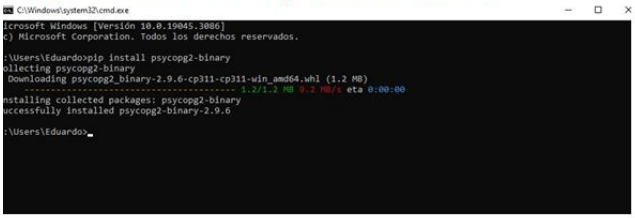
When carrying out this project, the opportunity was taken to put concepts from the classes into practice, such as the use of the Cloud, for which it was decided to create the database using Azure services, where each group participant had access to the same transformed dataset, run queries to understand the dataset and participate in the whole process being able to produce some insights.



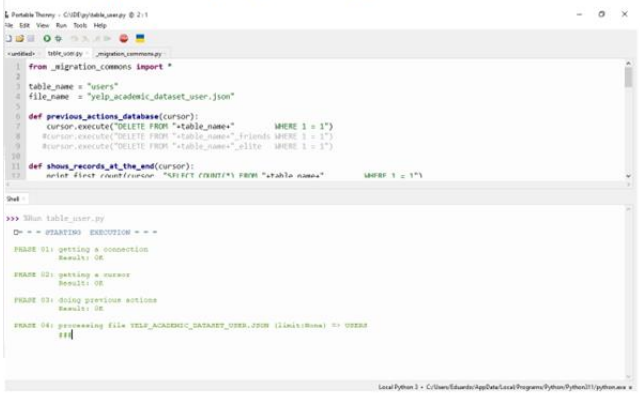
Dataset (json documents) imported to the Azure database.

Once the Python program and the database were ready, it was time to import the data to the Azure database through PgAdmin4. The following image shows the migration process for the table users:

PostgreSQL driver for Python installation

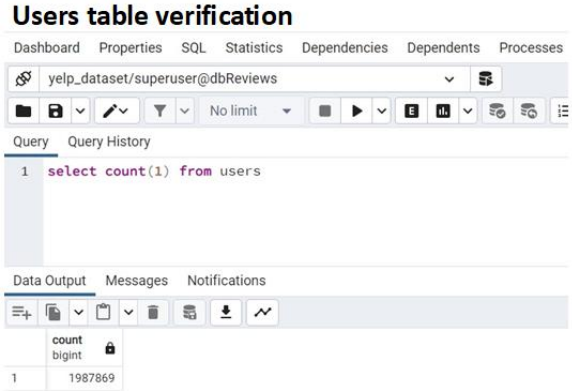


Running python program migration



Finishing process

Users table verification



During the dataset importation process to the Azure database, some issues were faced and solved, including character encoding problems, language errors, database connectivity, and other related

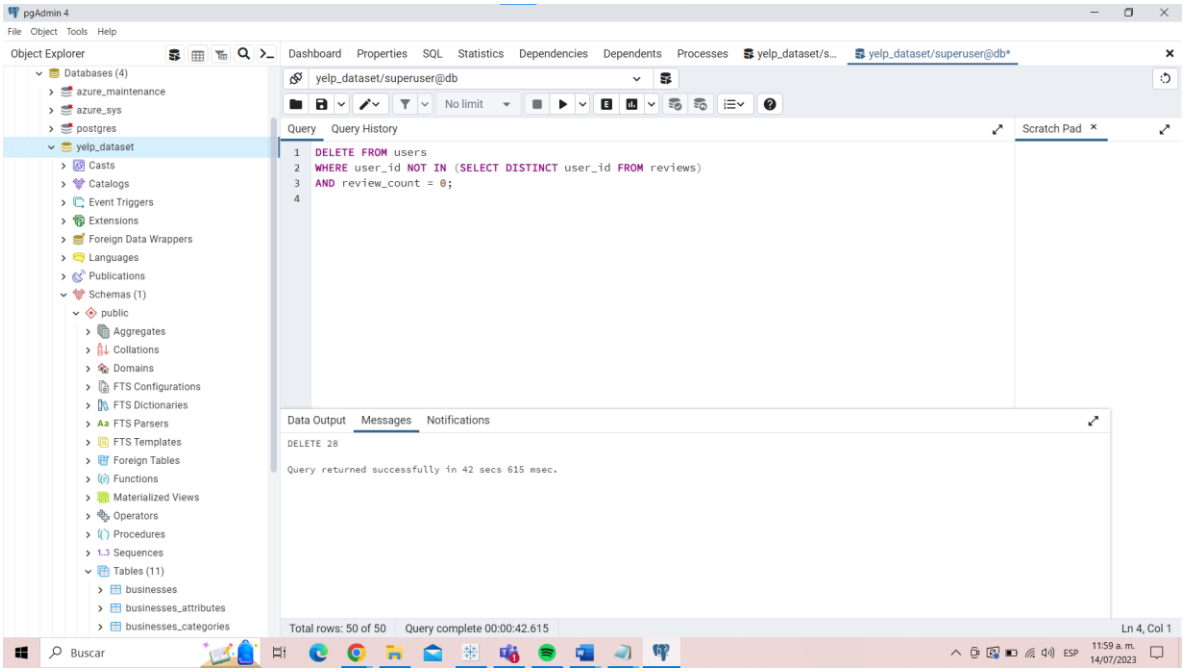
issues. It was a terrific opportunity to learn more about the cloud and how to deal with situations like these.

Dataset cleaning process

Specific queries were run during the cleaning phase to check for duplicate or inconsistent data and remove it from the analysis while maintaining the necessary data to have an accurate analysis and better results. The following queries were executed:

1. **Delete inactive users without reviews:** Inactive users with no reviews may be considered stale data or abandoned accounts. By removing them, metrics relating to active users become more accurate, and data analysis becomes easier.

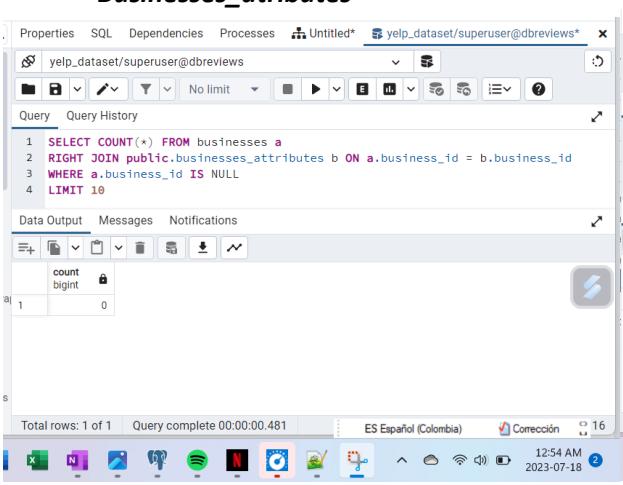
```
DELETE FROM users
WHERE user_id NOT IN (SELECT DISTINCT user_id FROM reviews)
AND review_count = 0;
```



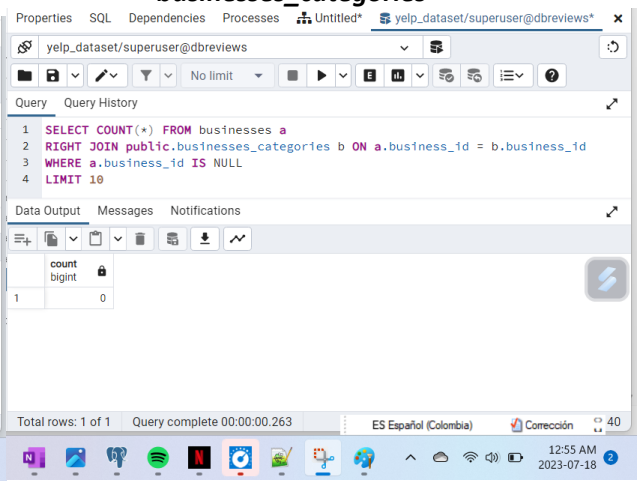
2. Identify records in attributes, categories, hours, reviews tables that are not associated to an id presented in the main table "business": This query is executed to analyse the consistency of the migrated dataset. The mentioned tables should not have records without a "business_id" that is not present in the main table "businesses". If this happens, it must be analysed to determine if it affected the integrity of the migrated data or if the dataset presents inconsistencies since all these features must be related to a business.

OUPUT:

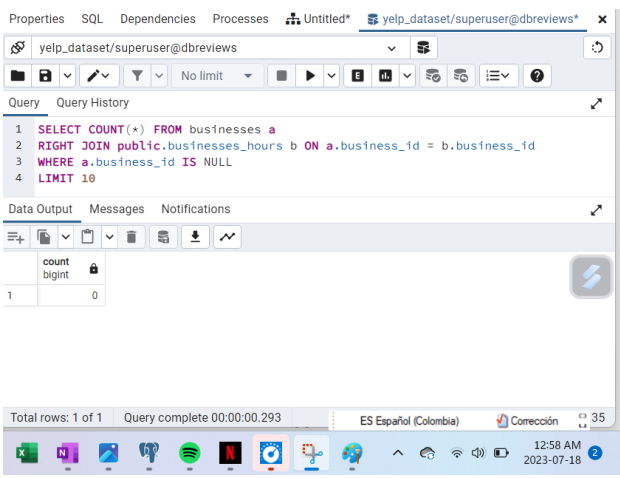
Businesses_atributes



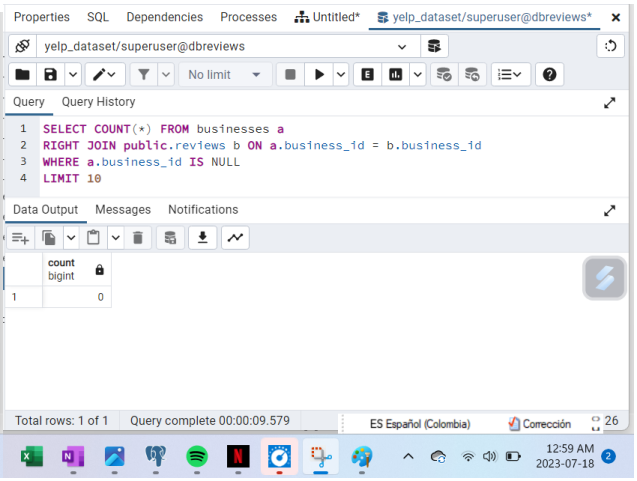
businesses_categories



Businesses_hours

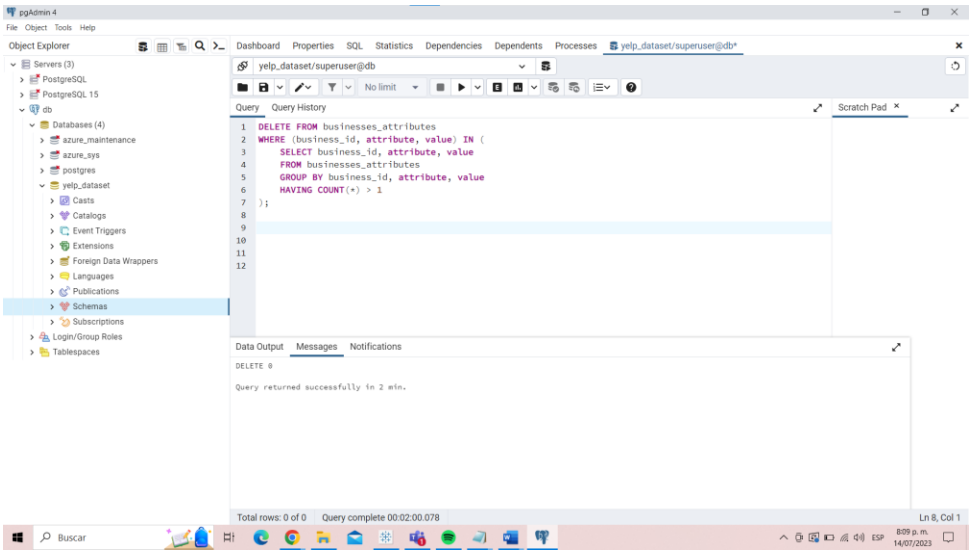


reviews

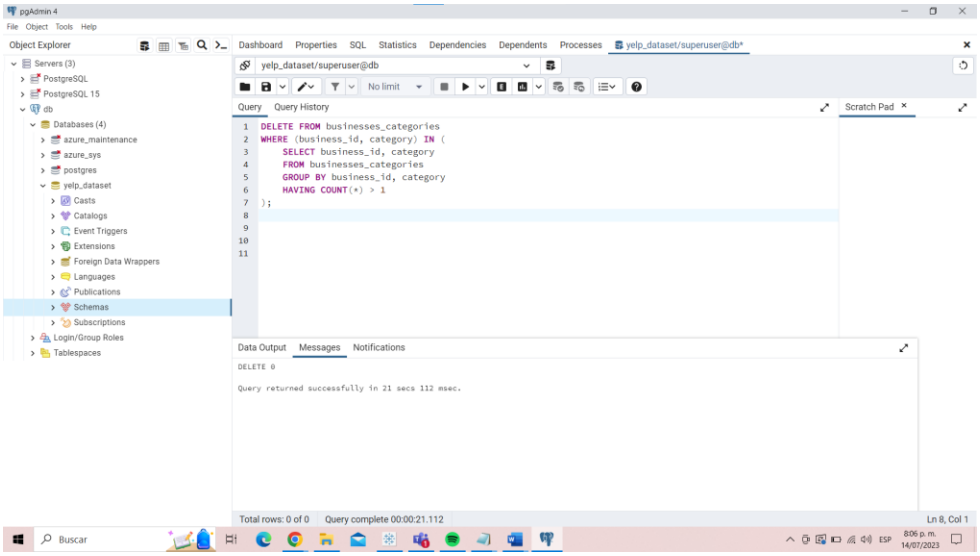


3. **Eliminate duplicate business attributes or business categories:** If there are duplicate records in the "businesses_attributes" or “businesses_categories” tables with the same combination of business, attribute, categories, and value, it may be redundant data or insert errors. Removing these duplicate attributes helps maintain consistency and prevents confusion when accessing data.

Query business_atributes:



Query businesses_categories:



After analysing the results of each last query and checking deeply the type and format of data, it was found to be a consistent database with a little "dirt" in just some fields. The information migrated is reliable and convenient for obtaining accurate results.

Dataset analysis and insights

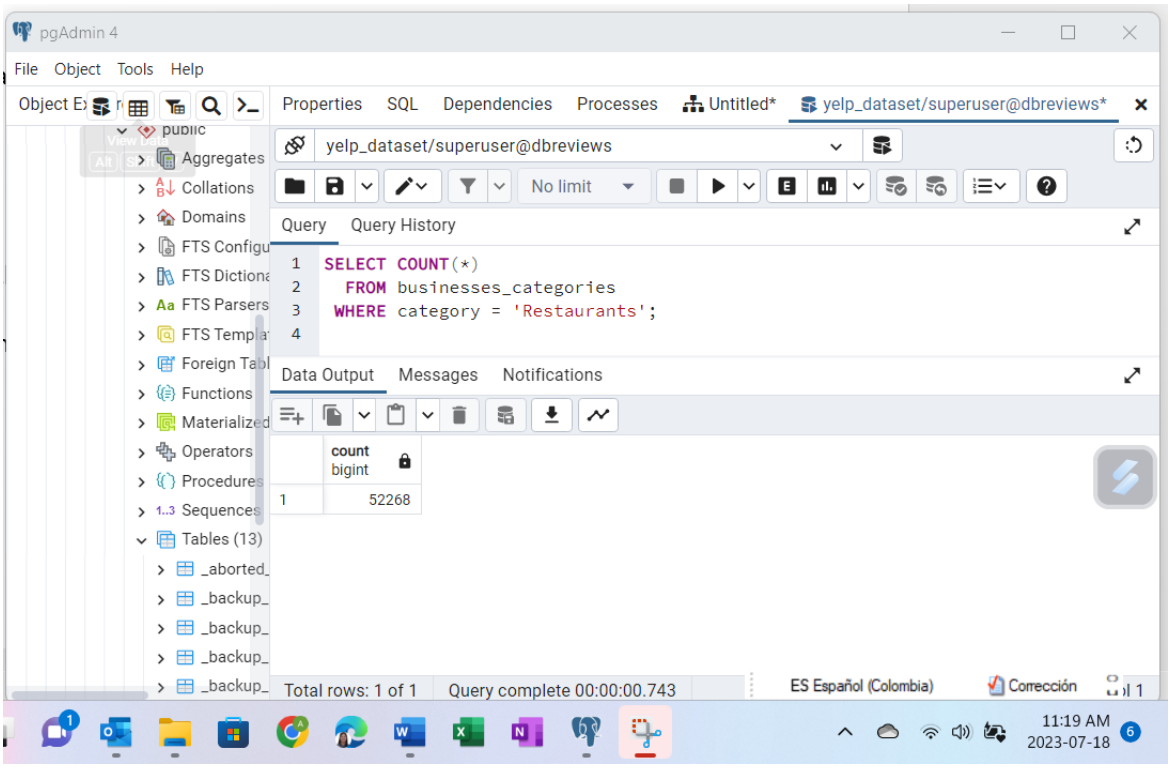
At this stage of the process, a brainstorming session was carried out with the team members to establish specific objectives to obtain significant information when analysing the data. In the first instance, general queries were made that led to the final queries that are shared at this stage of the documentation, with their respective descriptions.

For the data analysis, it was also established that the analysis would focus only on the businesses that are categorised as "Restaurants" in the database.

1. **Finding the number of businesses that are cataloged as restaurants:** it was executed a query to know the total number of restaurants in the database, since it is the category chosen for this project.

SQL:

```
SELECT COUNT(*)
 FROM businesses_categories
WHERE category = 'Restaurants';
```

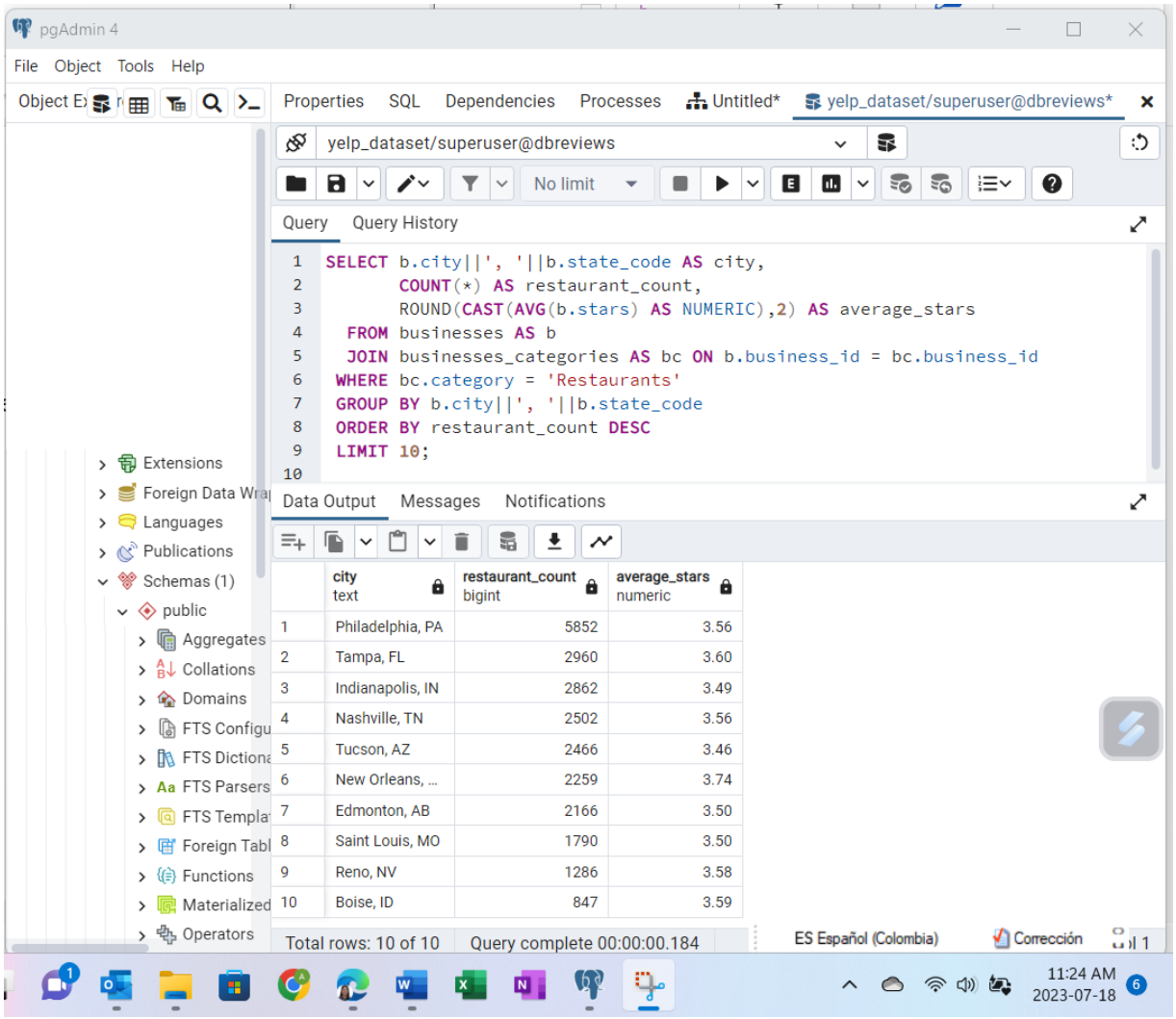


Results: A total of 52,268 businesses categorized as 'Restaurants' were found in the database.

2. **Find the 10 cities with the most restaurants and their respective average star rating.** With this query, the goal is to find out in which cities YELP has the largest number of registered restaurants and to have an approach to user satisfaction in those cities based on star ratings.

SQL:

```
SELECT b.city||', '||b.state_code AS city,
      COUNT(*) AS restaurant_count,
      ROUND(CAST(AVG(b.stars) AS NUMERIC),2) AS average_stars
 FROM businesses AS b
 JOIN businesses_categories AS bc ON b.business_id = bc.business_id
WHERE bc.category = 'Restaurants'
GROUP BY b.city||', '||b.state_code
ORDER BY restaurant_count DESC
LIMIT 10;
```



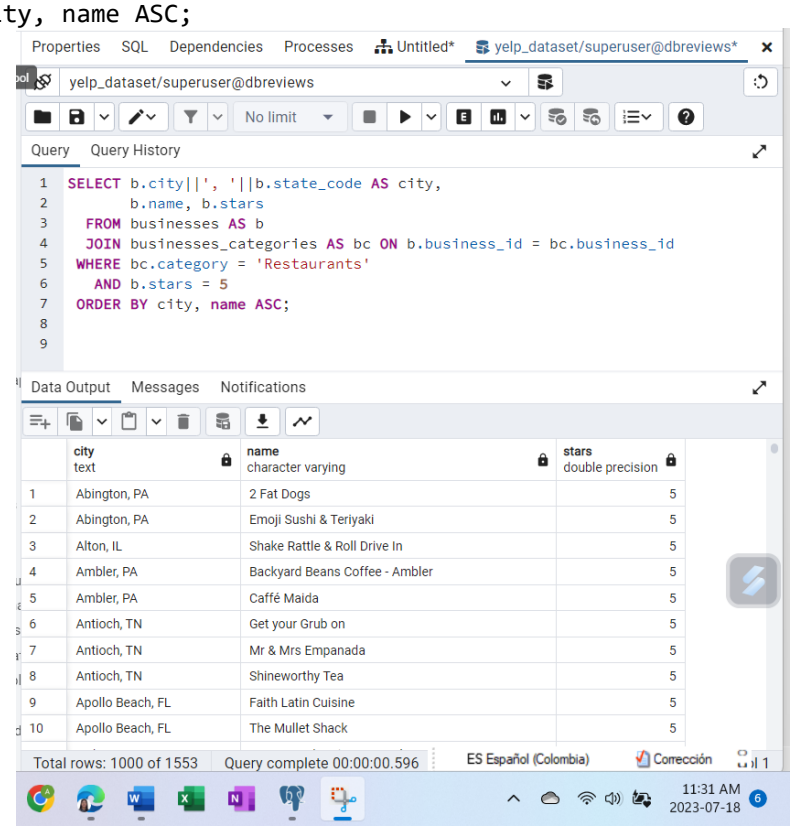
Results: After this query, it was identified that the city with the largest number of restaurants is Philadelphia with a total of 5,852 businesses in this category, with an approximate average of 3.56 stars. It was also found that 7 of the 10 cities with the largest number of restaurants have more than 2,000 business units of this type.

3. **Find the restaurants with the highest stars:** This query allows the restaurants with the highest rating (5 stars) to be identified and displayed on the front-end and promoted on the front page.

This helps highlight businesses that offer a great dining experience and attract potential customers looking for high-quality options.

SQL:

```
SELECT b.city||', '||b.state_code AS city,
       b.name, b.stars
  FROM businesses AS b
  JOIN businesses_categories AS bc ON b.business_id = bc.business_id
 WHERE bc.category = 'Restaurants'
   AND b.stars = 5
 ORDER BY city, name ASC;
```



	city text	name character varying	stars double precision
1	Abington, PA	2 Fat Dogs	5
2	Abington, PA	Emoji Sushi & Teriyaki	5
3	Alton, IL	Shake Rattle & Roll Drive In	5
4	Ambler, PA	Backyard Beans Coffee - Ambler	5
5	Ambler, PA	Caffé Maida	5
6	Antioch, TN	Get your Grub on	5
7	Antioch, TN	Mr & Mrs Empanada	5
8	Antioch, TN	Shineworthy Tea	5
9	Apollo Beach, FL	Faith Latin Cuisine	5
10	Apollo Beach, FL	The Mullet Shack	5

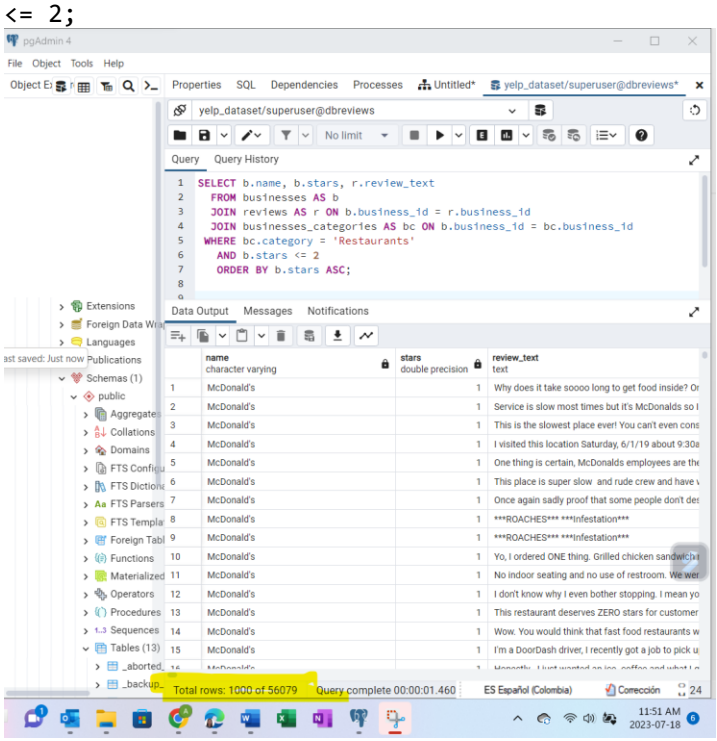
Results: After implementing the query, 1,553 restaurants were found with a high rating (5 stars) by their users.

4. **Identify businesses with the worst ratings (<=2) and analyze reviews to identify recurring issues:** By identifying the businesses with the worst ratings, Yelp can identify patterns and specific areas where businesses can improve. This may include aspects such as the quality of the products or services, customer service, cleanliness, waiting time, among others.

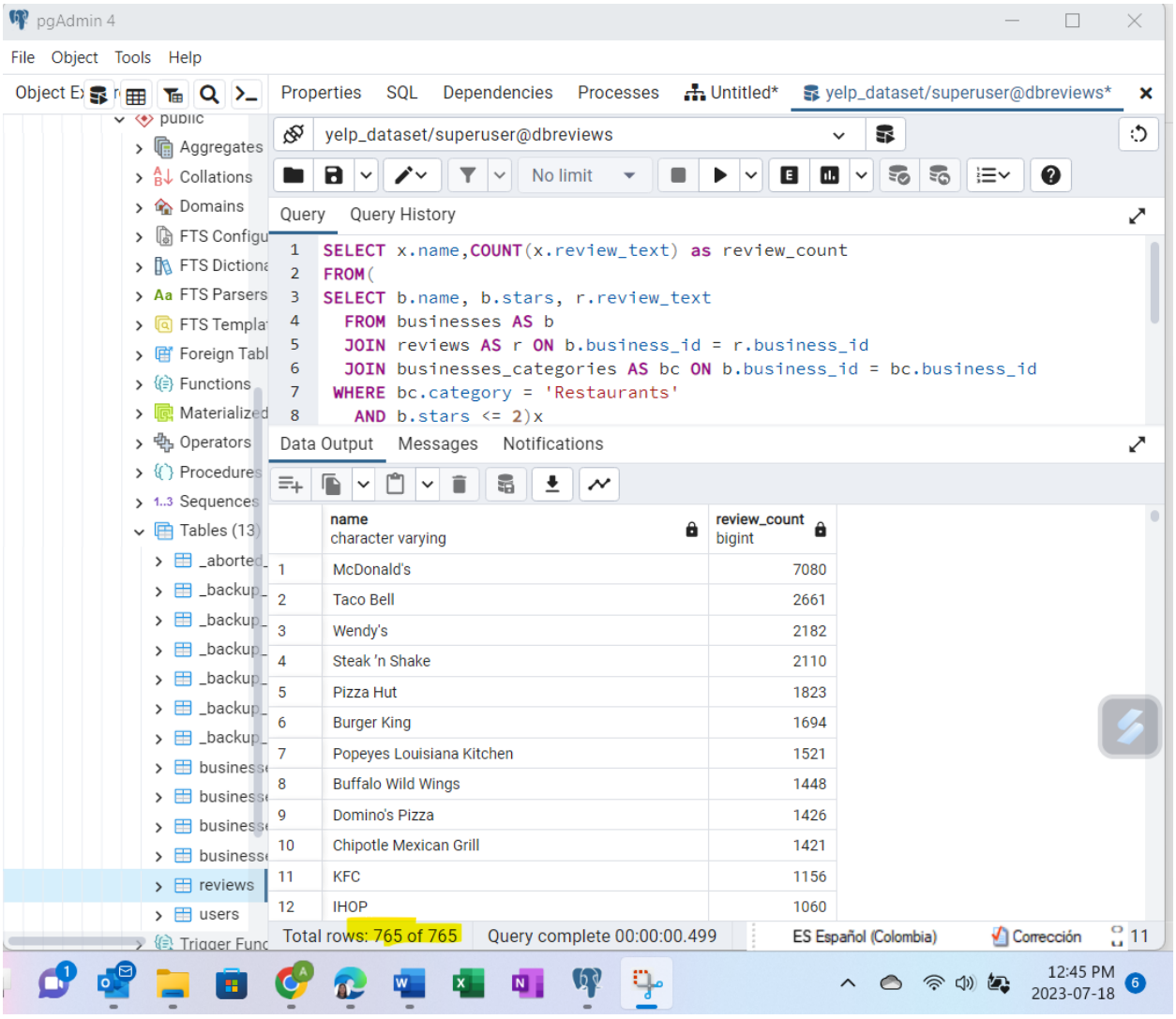
This analysis is important because the company is looking at a window of opportunity to expand and diversify the business to increase revenue, and this could help decide if the company could choose to offer a custom assessment to the lowest rated restaurant to identify weaknesses and advise them to improve and achieve better acceptance by users. This advice will be personalized and will be offered at a low cost to any restaurant that wishes to improve its reputation.

Query 1a:

```
SELECT b.name, b.stars, r.review_text
  FROM businesses AS b
  JOIN reviews AS r ON b.business_id = r.business_id
  JOIN businesses_categories AS bc ON b.business_id = bc.business_id
 WHERE bc.category = 'Restaurants'
   AND b.stars <= 2;
```



Query 1b: Identify the number of restaurants with the worst ratings (<=2)

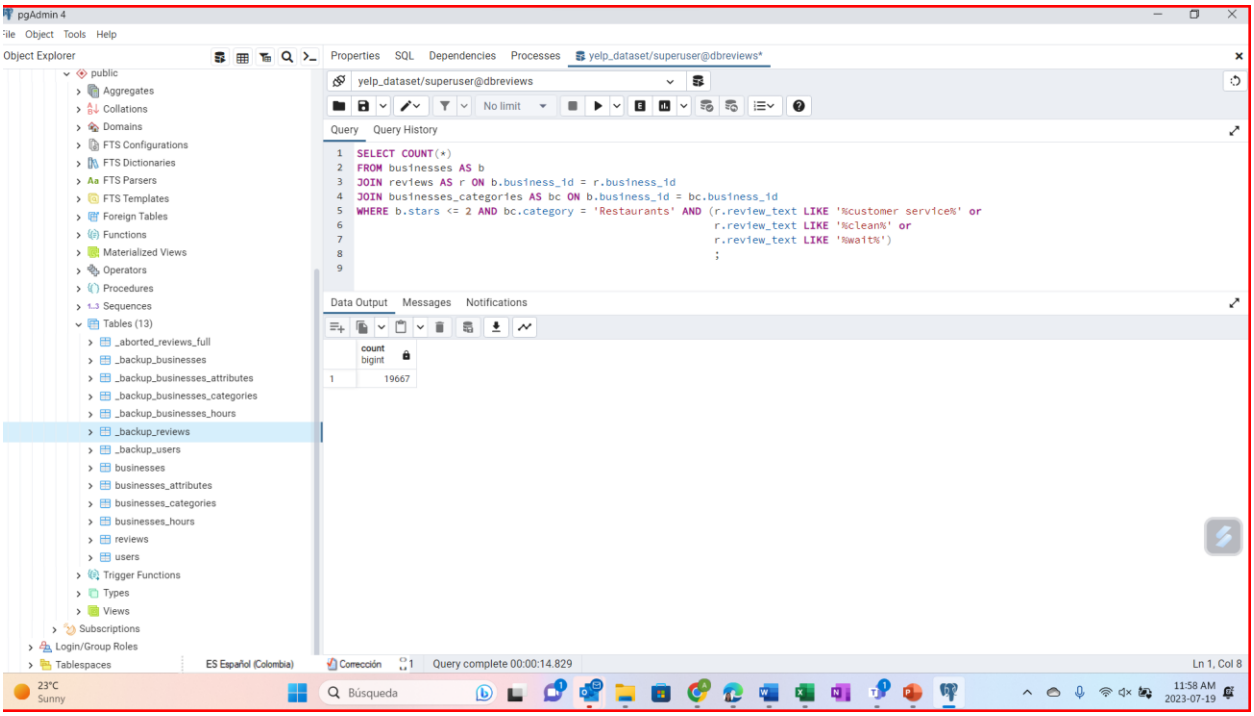


Result: 56,079 restaurant reviews were found with a star rating equal to or less than a score of 2. The top 3 of the restaurants with the worst reviews are MacDonald's, Taco Bell, and Wendy's.

To have more tools when analyzing the results of this query, it was executed three additional queries, filtering three keywords: "customer service", "clean" and "waiting time" grouping for restaurant and city and for those with more than 10 bad reviews.

Query 1c: Number of reviews related to the issues "customer service", "clean" or "waiting time"

```
SELECT COUNT(*)
FROM businesses AS b
JOIN reviews AS r ON b.business_id = r.business_id
JOIN businesses_categories AS bc ON b.business_id = bc.business_id
WHERE b.stars <= 2 AND bc.category = 'Restaurants' AND (r.review_text LIKE
'%customer service%' or r.review_text LIKE '%clean%' or r.review_text LIKE
'%wait%');
```

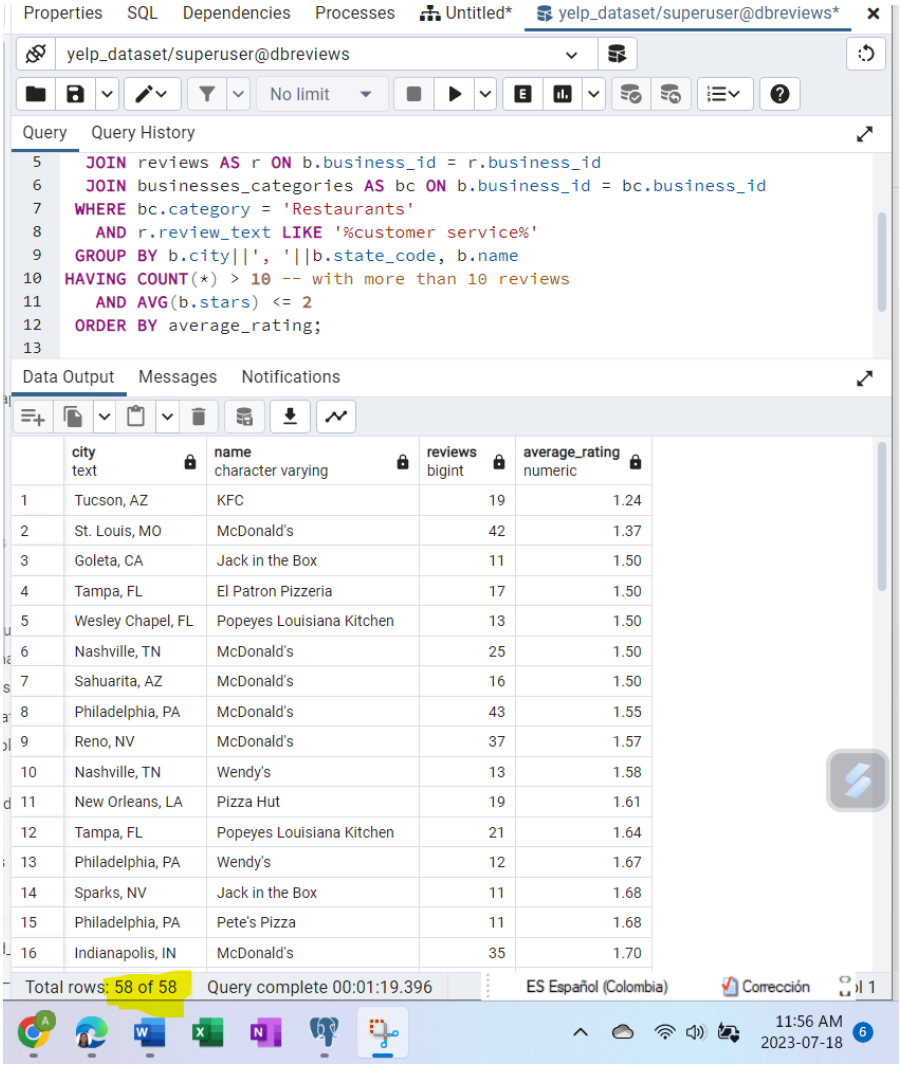


Result: 19,667 of 56,079 restaurant reviews that represents 35% were found with the issue "Customer services" or "Clean" or "Wait".

Query 2:

```
SELECT b.city||', '||b.state_code AS city, b.name,
       COUNT(*) reviews,
       ROUND(CAST(AVG(b.stars) AS NUMERIC),2) AS average_rating
  FROM businesses AS b
  JOIN reviews AS r ON b.business_id = r.business_id
  JOIN businesses_categories AS bc ON b.business_id = bc.business_id
 WHERE bc.category = 'Restaurants'
   AND r.review_text LIKE '%customer service%'
 GROUP BY b.city||', '||b.state_code, b.name
HAVING COUNT(*) > 10 -- with more than 10 reviews
   AND AVG(b.stars) <= 2
```

```
ORDER BY average_rating;
```

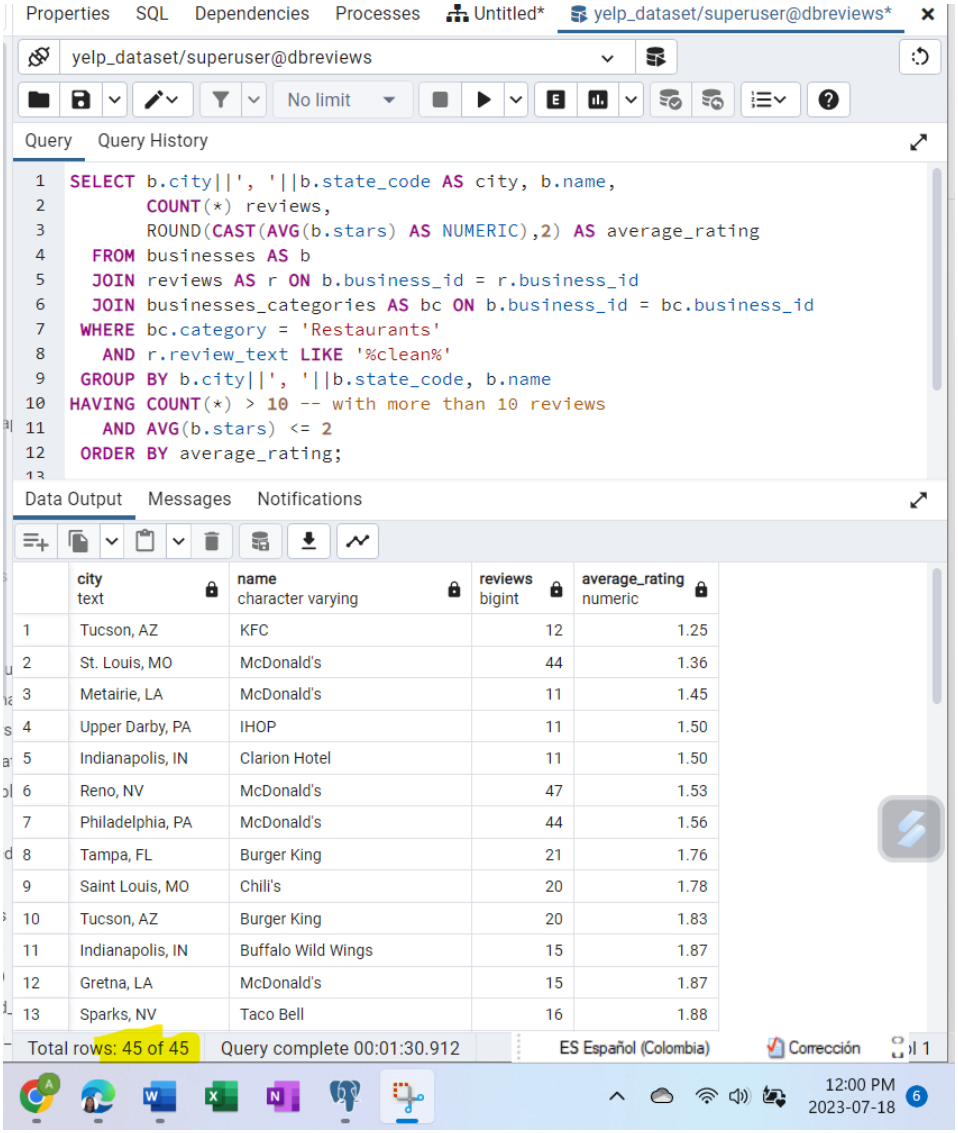


Results: 58 restaurants with a star rating less than or equal to 2 were found, where "customer service" is mentioned as at least 10 reviews.

Query 3:

```
SELECT b.city||', '||b.state_code AS city, b.name,
       COUNT(*) reviews,
       ROUND(CAST(AVG(b.stars) AS NUMERIC),2) AS average_rating
  FROM businesses AS b
  JOIN reviews AS r ON b.business_id = r.business_id
  JOIN businesses_categories AS bc ON b.business_id = bc.business_id
 WHERE bc.category = 'Restaurants'
   AND r.review_text LIKE '%clean%'
 GROUP BY b.city||', '||b.state_code, b.name
HAVING COUNT(*) > 10 -- with more than 10 reviews
   AND AVG(b.stars) <= 2
```

```
ORDER BY average_rating;
```

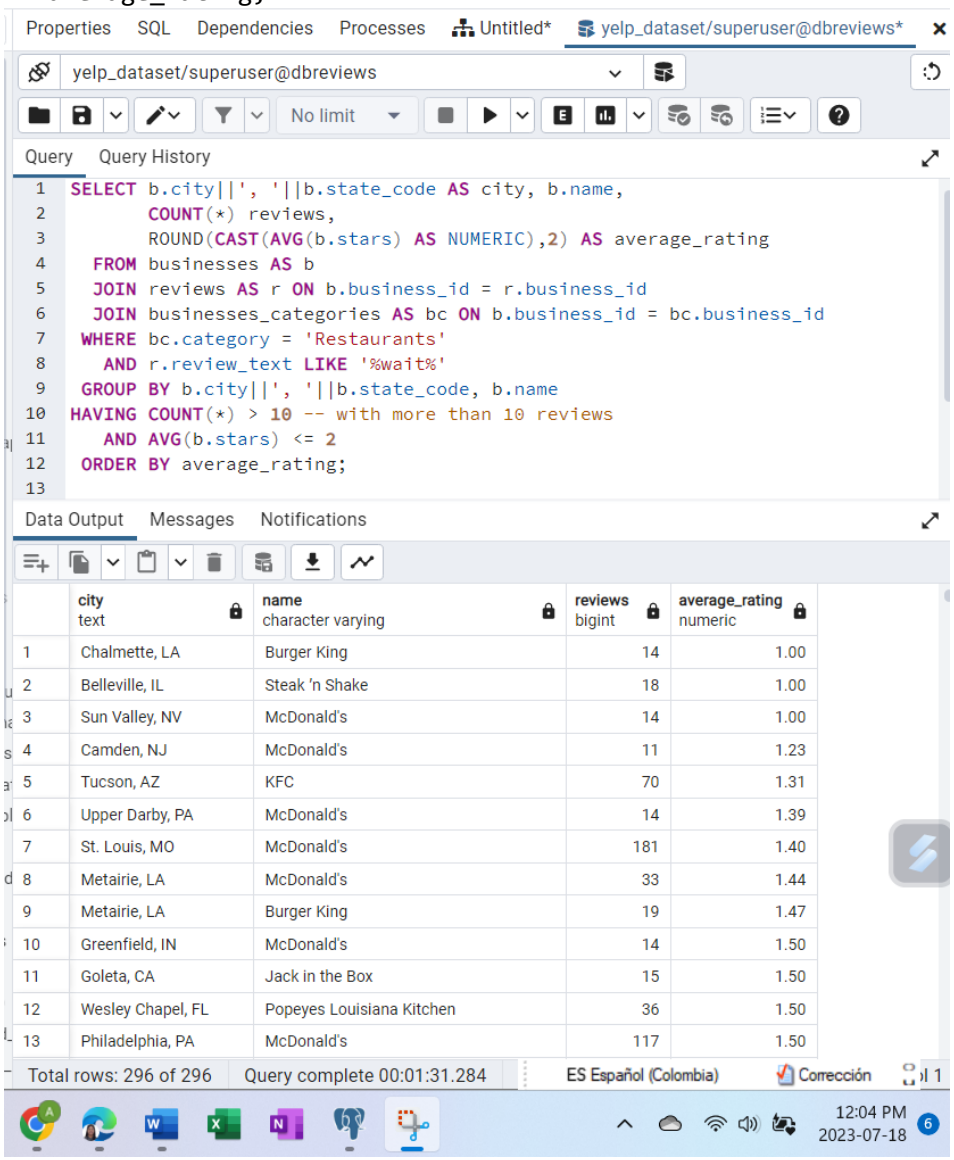


Result: 45 restaurants with a star rating less than or equal to 2 were found, where "clean" is mentioned as at least 10 reviews.

Query 4:

```
SELECT b.city||', '||b.state_code AS city, b.name,
       COUNT(*) reviews,
       ROUND(CAST(AVG(b.stars) AS NUMERIC),2) AS average_rating
  FROM businesses AS b
  JOIN reviews AS r ON b.business_id = r.business_id
  JOIN businesses_categories AS bc ON b.business_id = bc.business_id
 WHERE bc.category = 'Restaurants'
   AND r.review_text LIKE '%wait%'
 GROUP BY b.city||', '||b.state_code, b.name
```

```
                                    HAVING COUNT(*) > 10 -- with more than 10
                                    reviews
  AND AVG(b.stars) <= 2
ORDER BY average_rating;
```



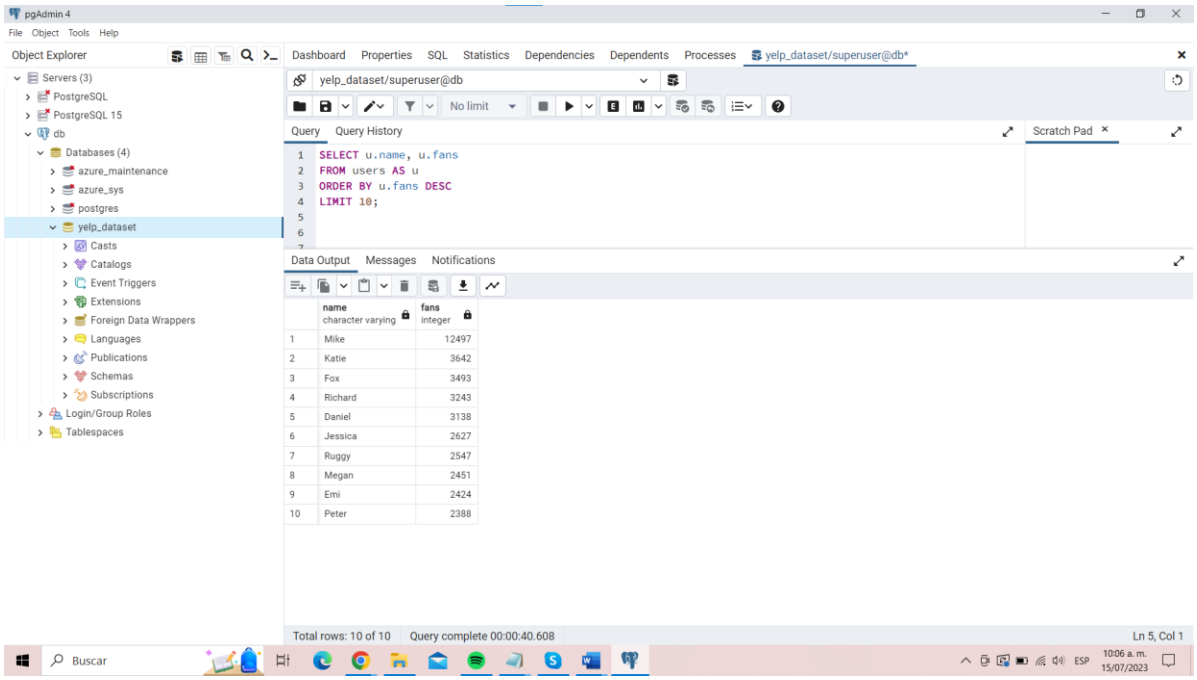
Result: 296 restaurants with a star rating less than or equal to 2 were found, where "wait" is mentioned as at least 10 reviews.

5. Find the most influential users (largest number of followers) and collaborate with them to promote businesses: By identifying the most influential users, Yelp can establish strategic collaborations with them to promote their services and highlight certain businesses or events. These

influential users can have a significant impact on the opinion and decisions of their followers, which could increase Yelp's visibility and reputation.

SQL:

```
SELECT u.name, u.fans
FROM users AS u
ORDER BY u.fans DESC
LIMIT 10;
```



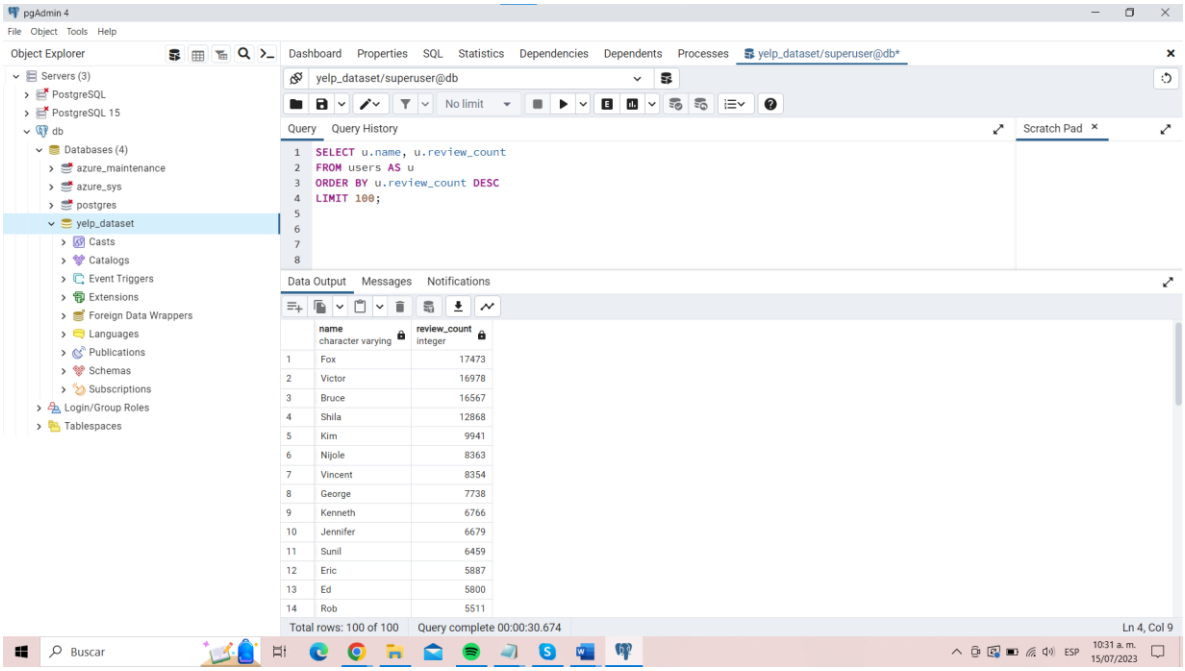
Result: The 10 users with the largest number of fans were found, among these the one with the most followers has a total of 12497, approximately 29% more than the user with the second most fans. All these users have more than 2,000 followers, a significant number of fans.

6. **Find the most active users and reward them with special promotions to build their loyalty:**

Knowing the most active users, Yelp can create customer loyalty strategies for them, such as rewarding these users with special promotions, thus strengthening the loyalty of those users towards the platform. By receiving exclusive incentives, these users can feel valued and recognized, which increases their satisfaction and motivates them to continue using Yelp as their preferred platform for finding and reviewing businesses.

SQL:

```
SELECT u.name, u.review_count
FROM users AS u
ORDER BY u.review_count DESC
LIMIT 100;
```



Return: This was the result obtained for the 100 users with the largest number of reviews, the first of which is called Fox and has 17,473 reviews.

After implementing the improvements in its services, YELP will be able to recommend the most active users visit these restaurants and rate them in order to improve their score and start a loyalty programme where these same people attract more people.

Spark

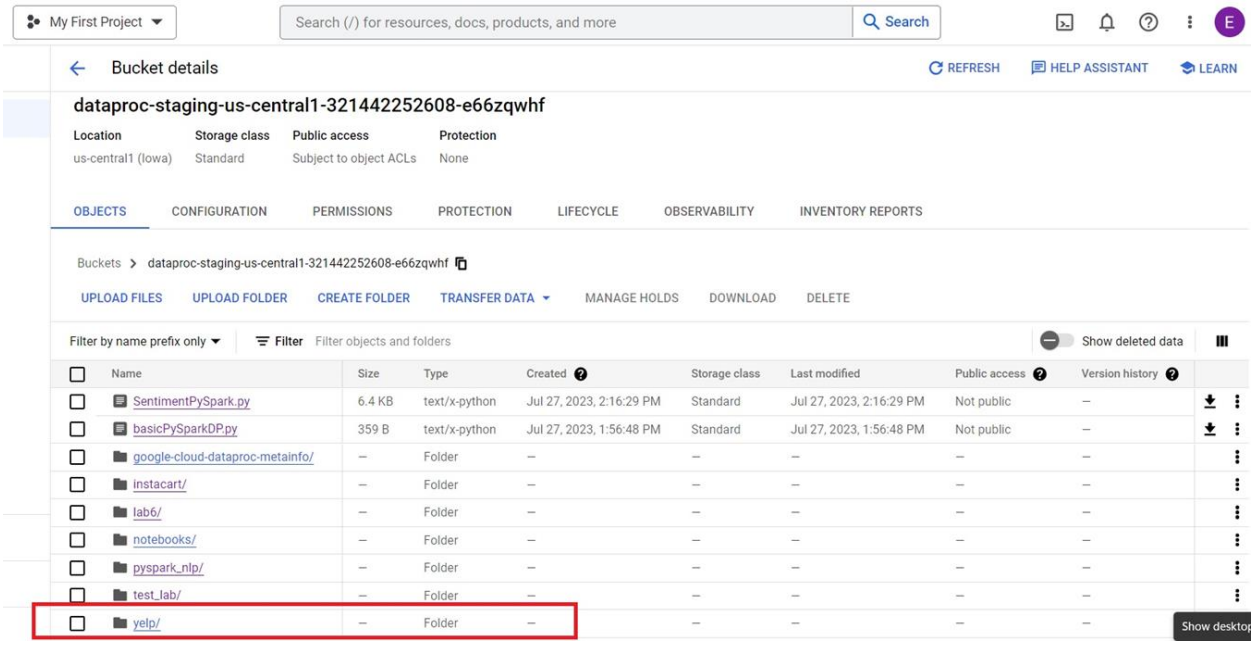
Running some jobs using PySpark

This section will provide a summarized report of the PySpark jobs that were run for this project, The purpose of this section is to document the results of running PySpark jobs on our set of CSV files. The data in the CSV files consists of the cleaned dataset as of the previous step's output, and the PySpark jobs' goal was to analyze the data and identify trends.

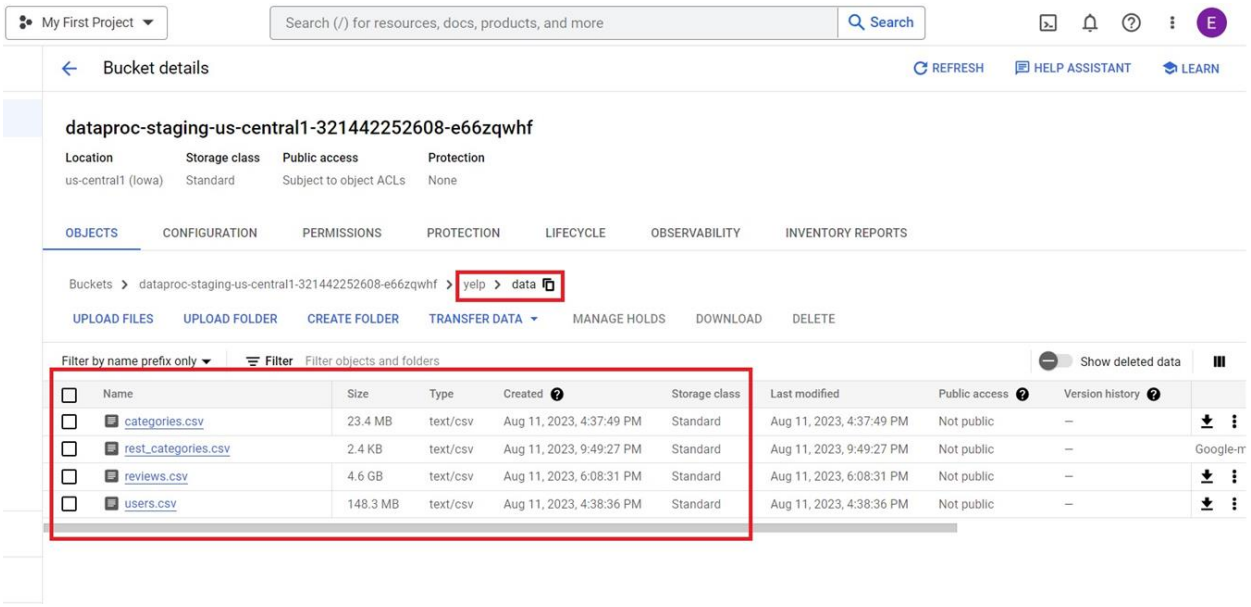
Methodology

The following steps were taken to run the PySpark jobs:

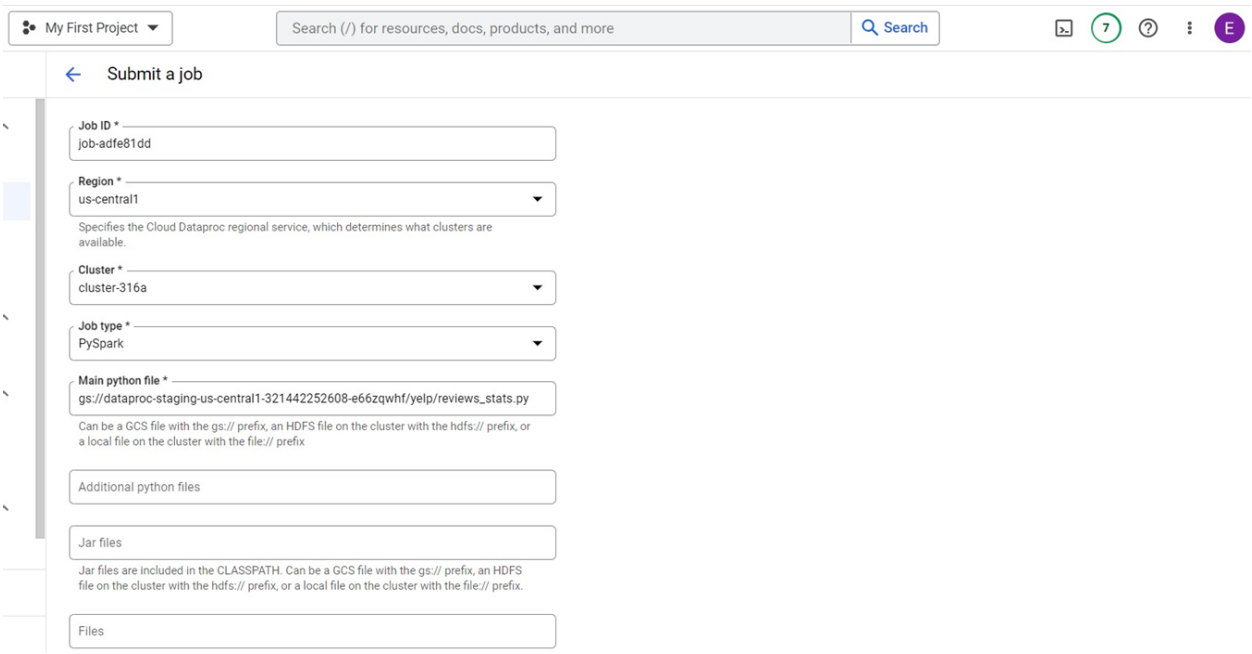
Firstly A specific working folder was created.

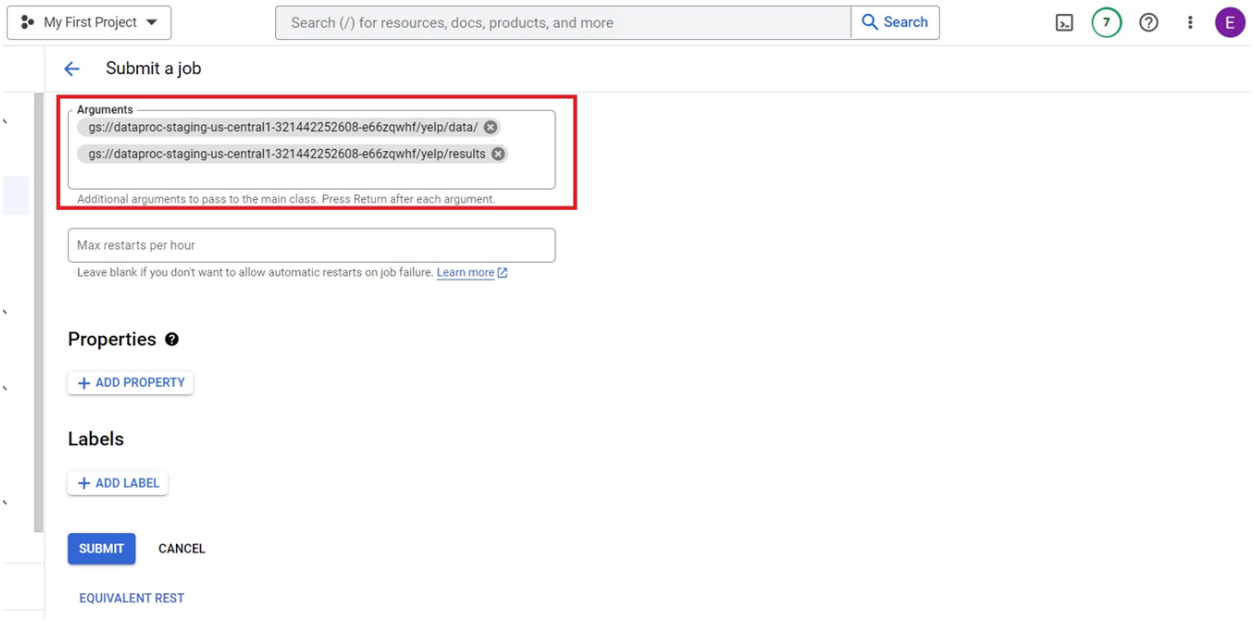


For the second step, The CSV files were uploaded to the working folder.

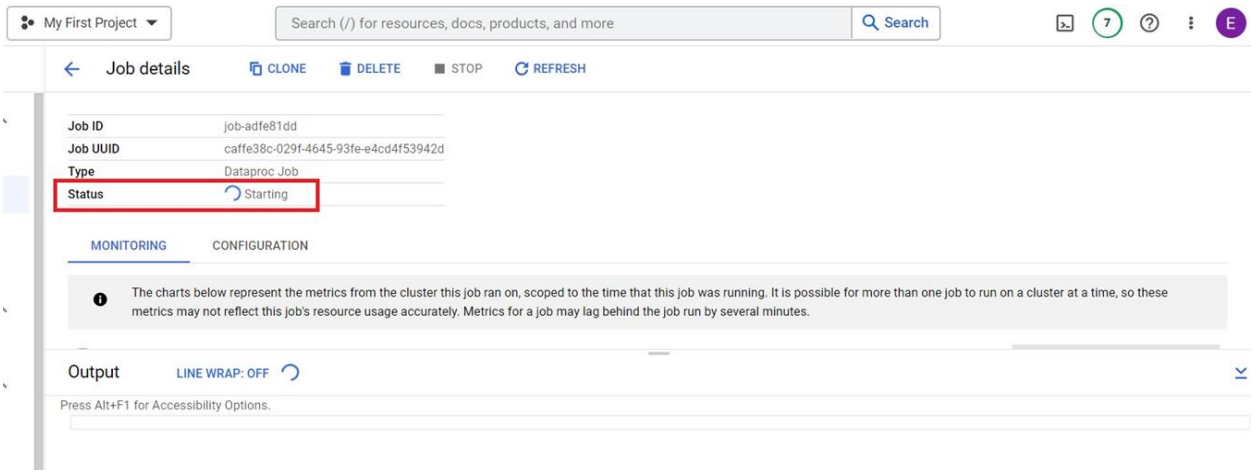


And next, A PySpark job was created to process the CSV files.



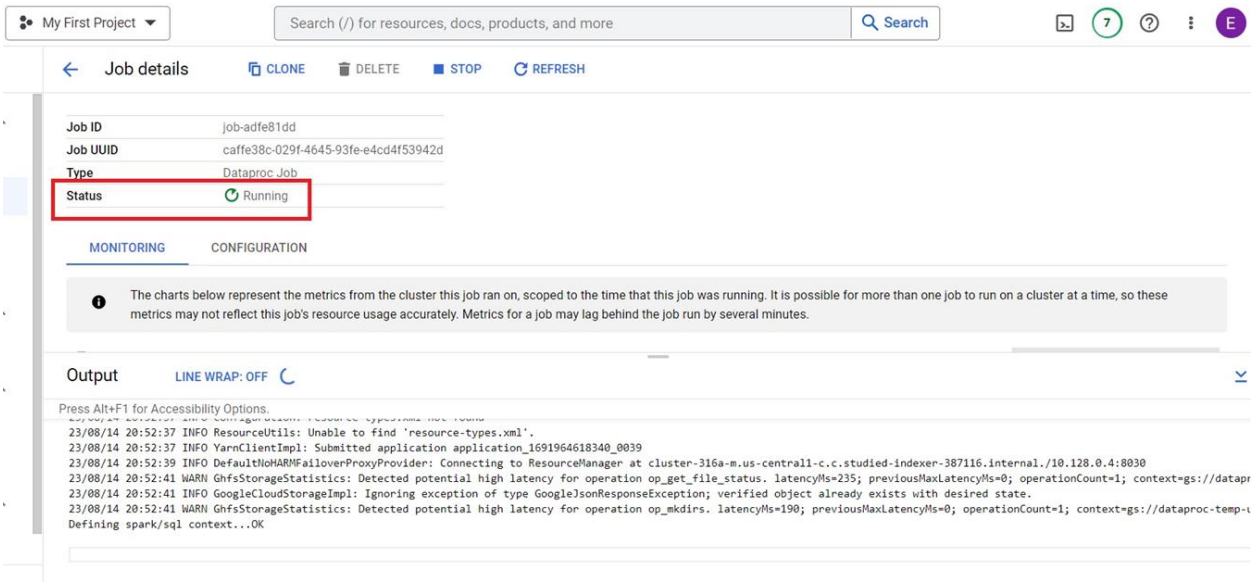


After that, The PySpark job was launched.

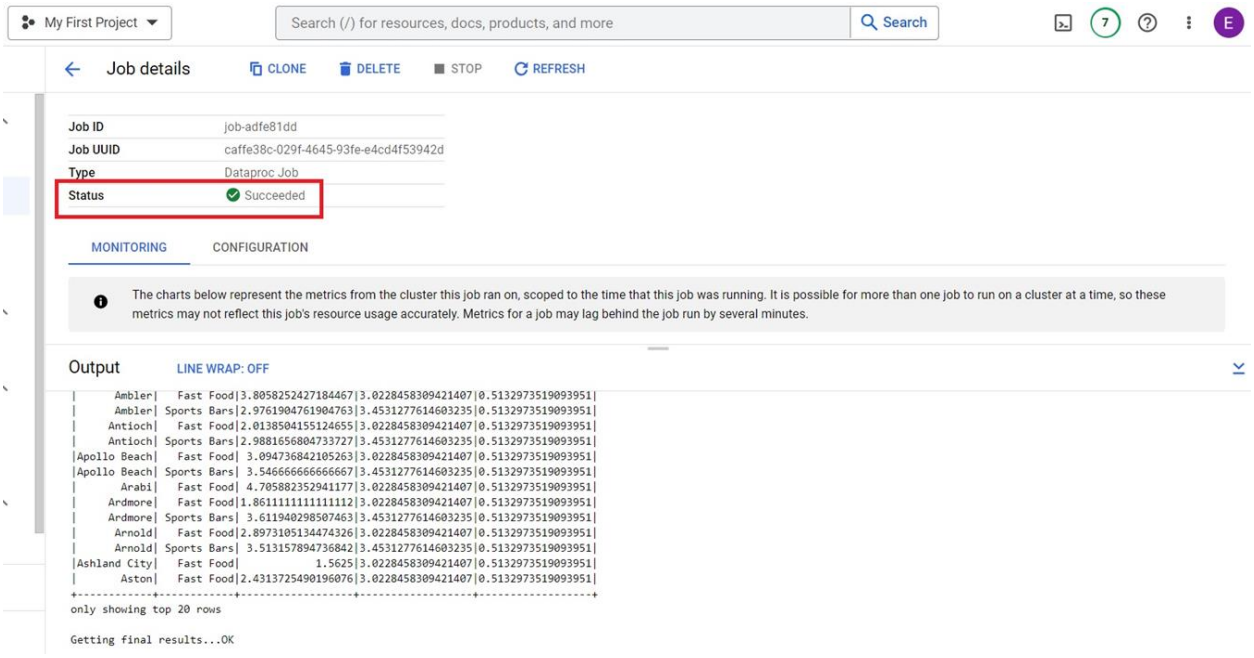


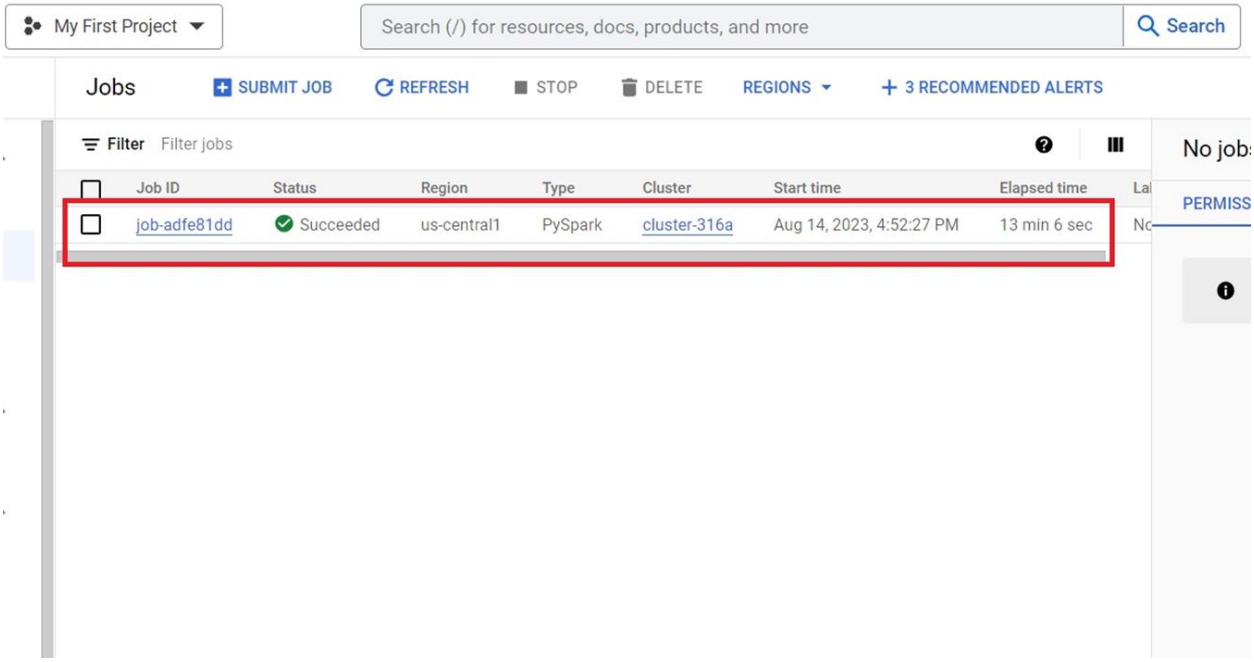
The PySpark job was monitored until it was completed successfully.

Job is running

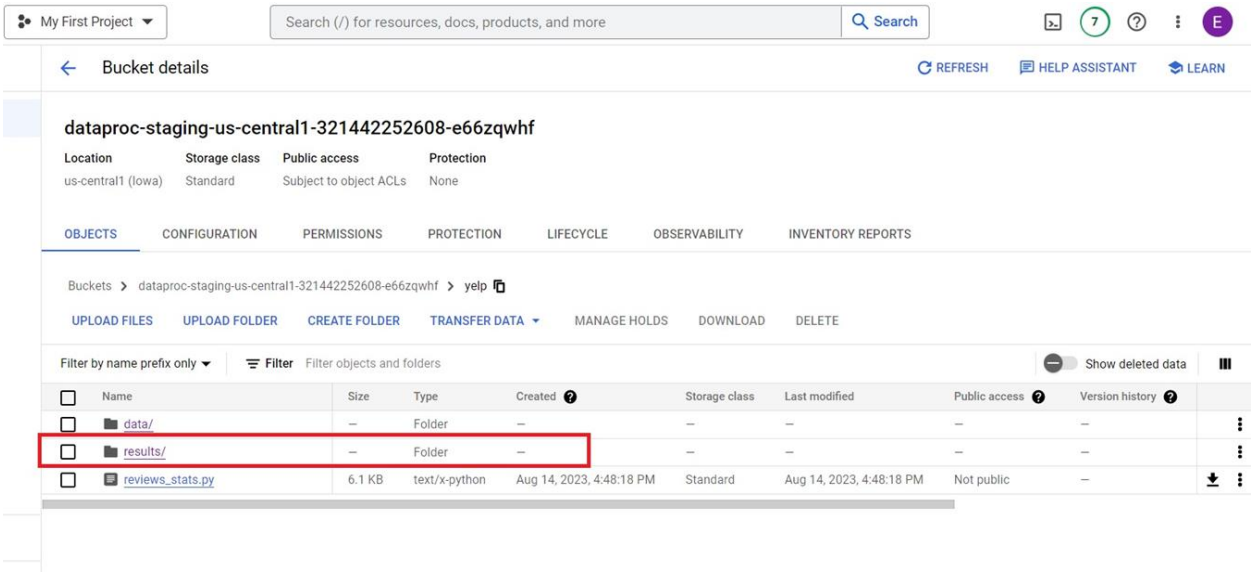


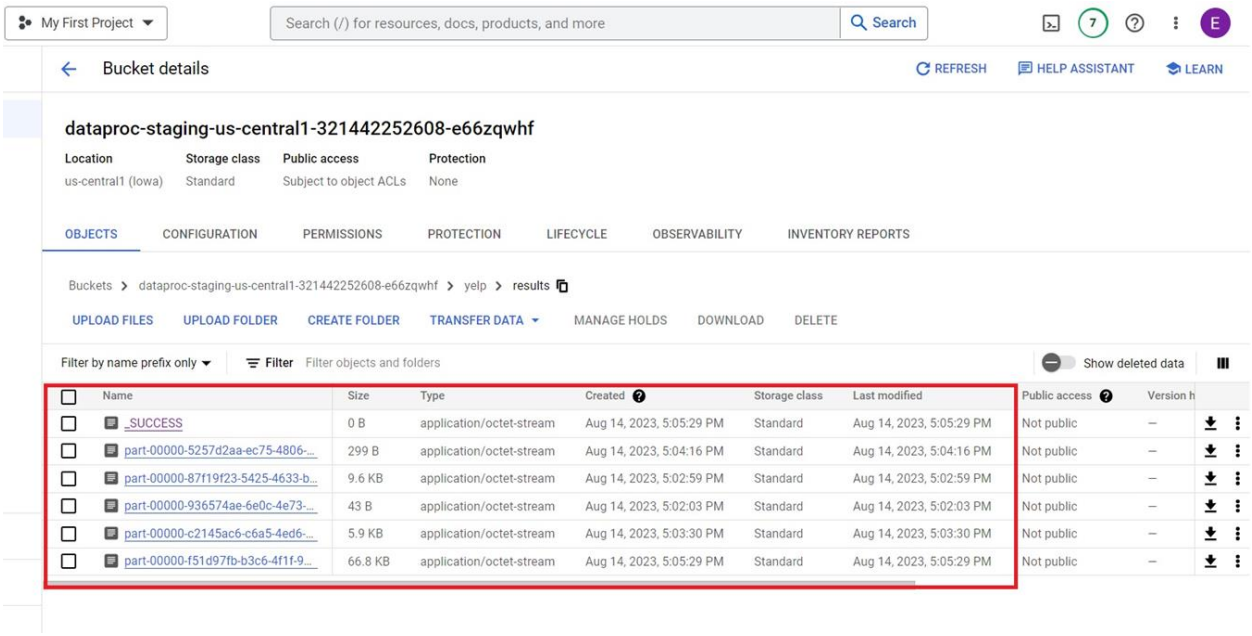
Job successfully completed





The results of the PySpark job were checked out.





The PySpark jobs were successfully done, and the outputs will present the results of the PySpark jobs for use in the next Step: Visualization

The PySpark jobs that were run produced five output files:

- `yelp_results_part-00000-936574ae-6e0c-4e73-80ba-813bc97c8824-c000.csv: (qty_categories.csv)`

  The output file contains the number of categories that are above the general average rating, whereas the number of categories that are below the general average rating.

- `yelp_results_part-00000-87f19f23-5425-4633-ba44-e0c26156f3d3-c000.csv: (avg_categories.csv)`

  The output file contains the average rating by category, with the general average rating in the last column.

- `yelp_results_part-00000-5257d2aa-ec75-4806-9274-8eee9dff1d56-c000.csv:`

  The output file contains the ten best and worst-rated categories, with the best-rated categories listed first.

- `yelp_results_part-00000-c2145ac6-c6a5-4ed6-acb6-3be86e8c2a10-c000.csv: (standard_deviation.csv)`

  The output file contains the standard deviation of the ratings by category.

- `yelp_results_part-00000-f51d97fb-b3c6-4f1f-9409-16b7c9a809cc-c000.csv:`

  This file contains the average rating by city and category, with the correlation coefficient between the average rating and the general average rating in the last column.

The PySpark jobs provided a lot of interesting insights into the Yelp dataset. The findings from the jobs can be used to inform future marketing and business decisions for businesses on Yelp. For example, businesses in the "Restaurants" category may want to focus on improving their customer service and food quality in order to maintain their high average rating.

The related code for running PySpark Jobs and the outputs file are uploaded to the project git repository. Please check it for any further information.

Findings from the result it would be more understandable by looking at graphs and charts. Let's do the Visualizations Part

Visualization

Technologies:

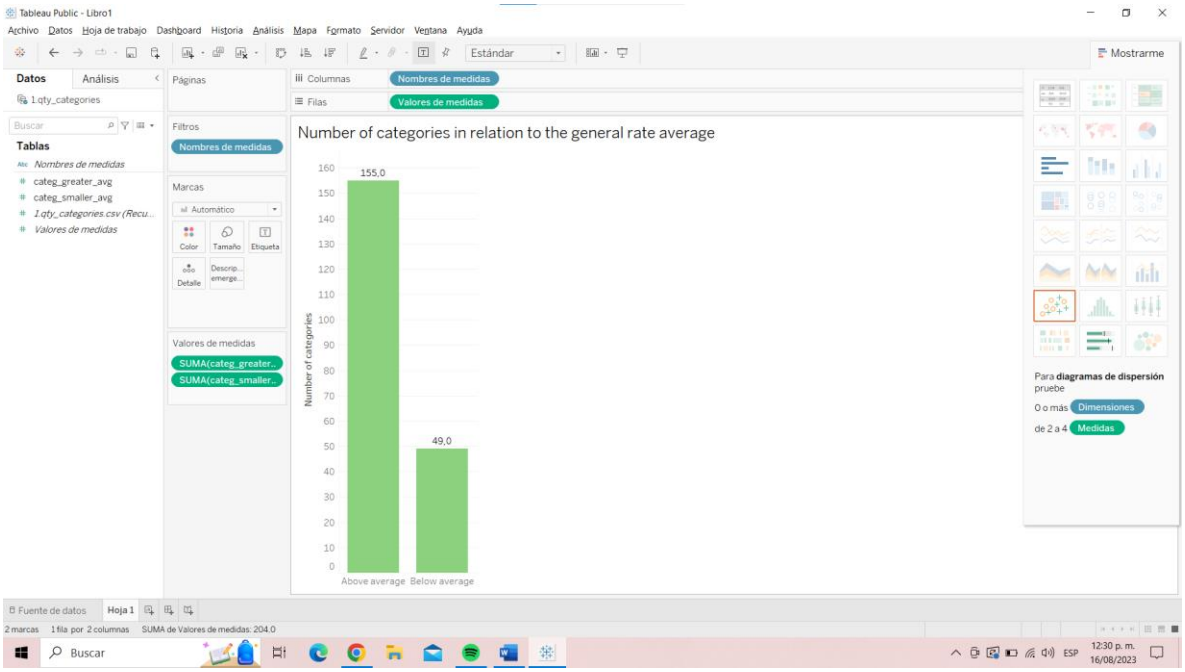
We chose Tableau for our project due to its ability to create interactive visualizations, effective presentations, and data-driven analysis through a wide range of visualizations and data connections, thereby enhancing our informed decision-making.

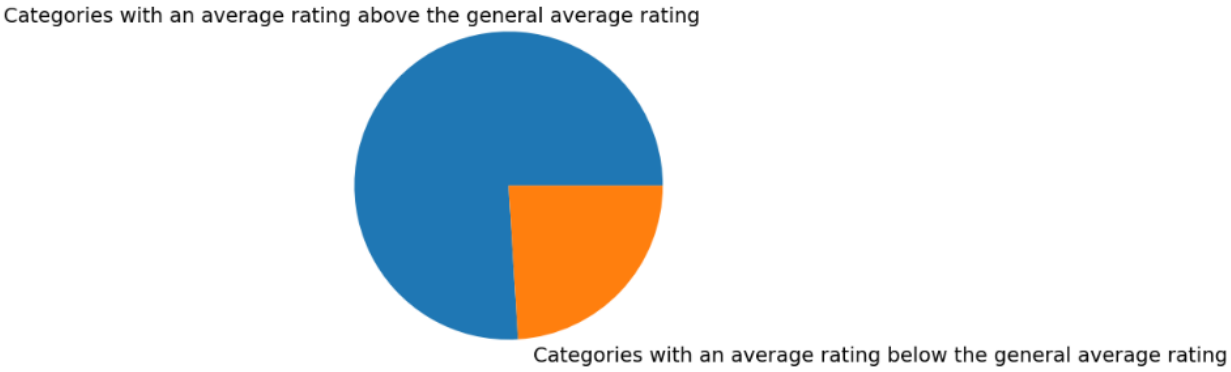
Also using Matplotlib due the power of ease and quick plot generation

The following are some of the key findings from the PySpark jobs based on the extracted graphs:

Number of categories in relation to the general rate average.

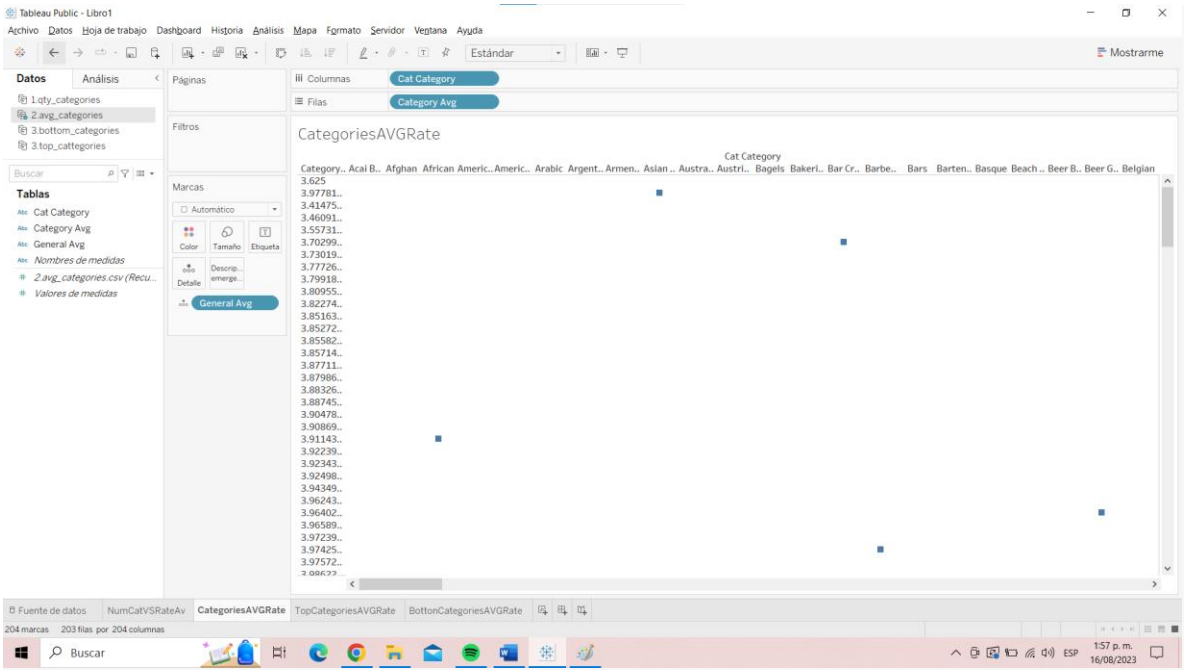
The two graphs reveal that the distribution of restaurant categories is uneven in relation to their average ratings. A significant majority of categories (155) are rated above average, while a much smaller number (49) are rated below average. This indicates that there is a wide range of restaurant quality, with many restaurants performing well above average and a few performing below average.





Average Rates of Each Category

At this point of view we wanted to find the average rates for each of the restaurant categories and compare it to the general average.



However, as can be seen in the visualization, since there is so much data (204) the statistics cannot be clearly observed, for this reason we decided to divide the table into two new tables, to be able to graph the top 5 (above the general average of rates) and the bottom 5 (below the general rate average) of the rate averages by category and to be able to observe them more clearly, these are the results:

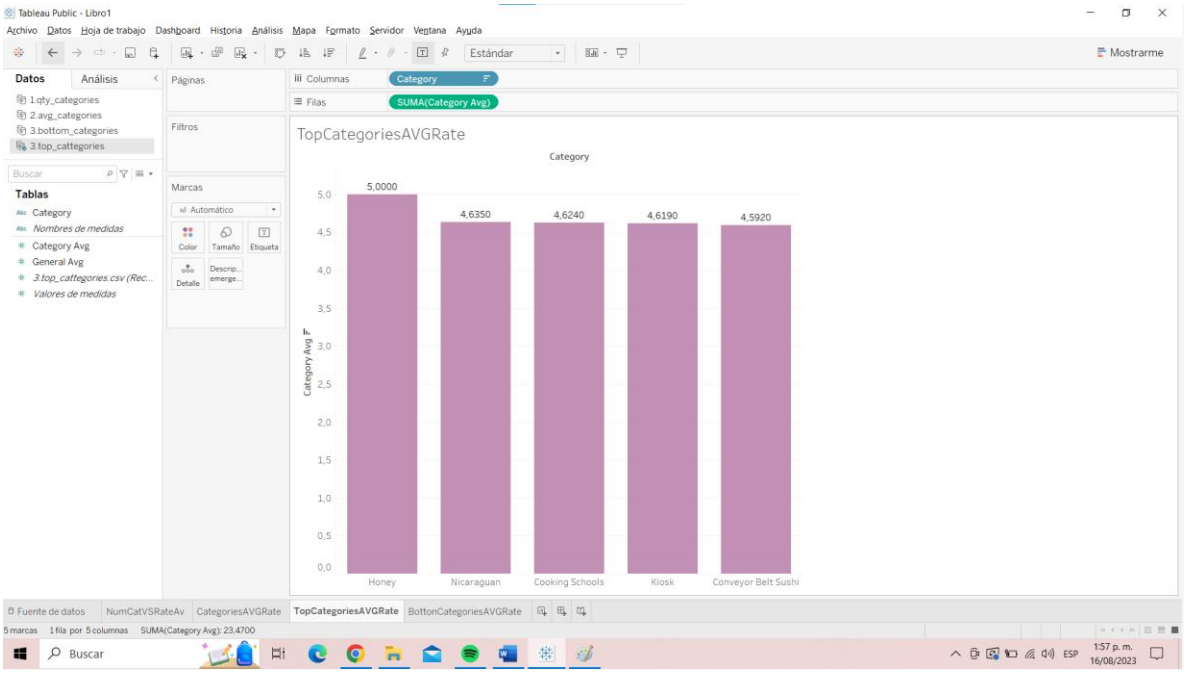


Tableau Public - Libro1
Archivo Datos Hoja de trabajo Dashboard Historia Análisis Mapa Formato Servidor Ventana Ayuda
Estándar
Mostrarme
Datos
Análisis
1.qty_categories
2.avg_categories
3.bottom_categories
3.top_cattegories
Páginas
Filtros
Marcas
Automático
Color
Tamaño
Etiqueta
Detalle
Tablas
Category
Nombres de medidas
Category Avg
General Avg
3.top_cattegories.csv (Rec...
Valores de medidas
Columnas
Category
Filas
SUMA(Category Avg)
TopCategoriesAVGRate
Category
Category Avg F
5,0000
4,6350
4,6240
4,6190
4,5920
Honey
Nicaraguan
Cooking Schools
Kiosk
Conveyor Belt Sushi
Fuente de datos
NumCatVSRateAv
CategoriesAVGRate
TopCategoriesAVGRate
BottonCategoriesAVGRate
5 marcas 1 fila por 5 columnas SUMA(Category Avg): 23.4700
Buscar
ESP
1:57 p. m.
16/08/2023

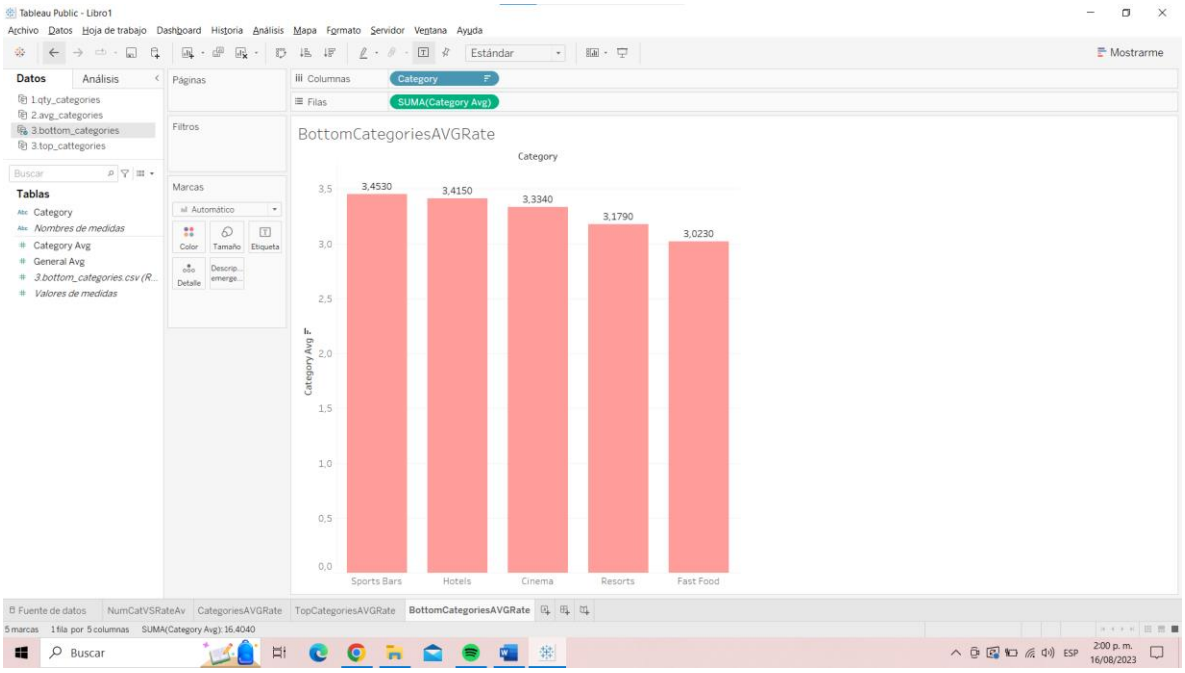


Tableau Public - Libro1
Archivo Datos Hoja de trabajo Dashboard Historia Análisis Mapa Formato Servidor Ventana Ayuda
Estándar
Mostrarme
Datos
Análisis
1.qty_categories
2.avg_categories
3.bottom_categories
3.top_cattegories
Páginas
Filtros
Marcas
Automático
Color
Tamaño
Etiqueta
Detalle
Tablas
Category
Nombres de medidas
Category Avg
General Avg
3.bottom_categories.csv (R...
Valores de medidas
Columnas
Category
Filas
SUMA(Category Avg)
BottomCategoriesAVGRate
Category
Category Avg F
3,4530
3,4150
3,3340
3,1790
3,0230
Sports Bars
Hotels
Cinema
Resorts
Fast Food
Fuente de datos
NumCatVSRateAv
CategoriesAVGRate
TopCategoriesAVGRate
BottomCategoriesAVGRate
5 marcas 1 fila por 5 columnas SUMA(Category Avg): 16.4040
Buscar
ESP
2:00 p. m.
16/08/2023

Average Rates of Each Category

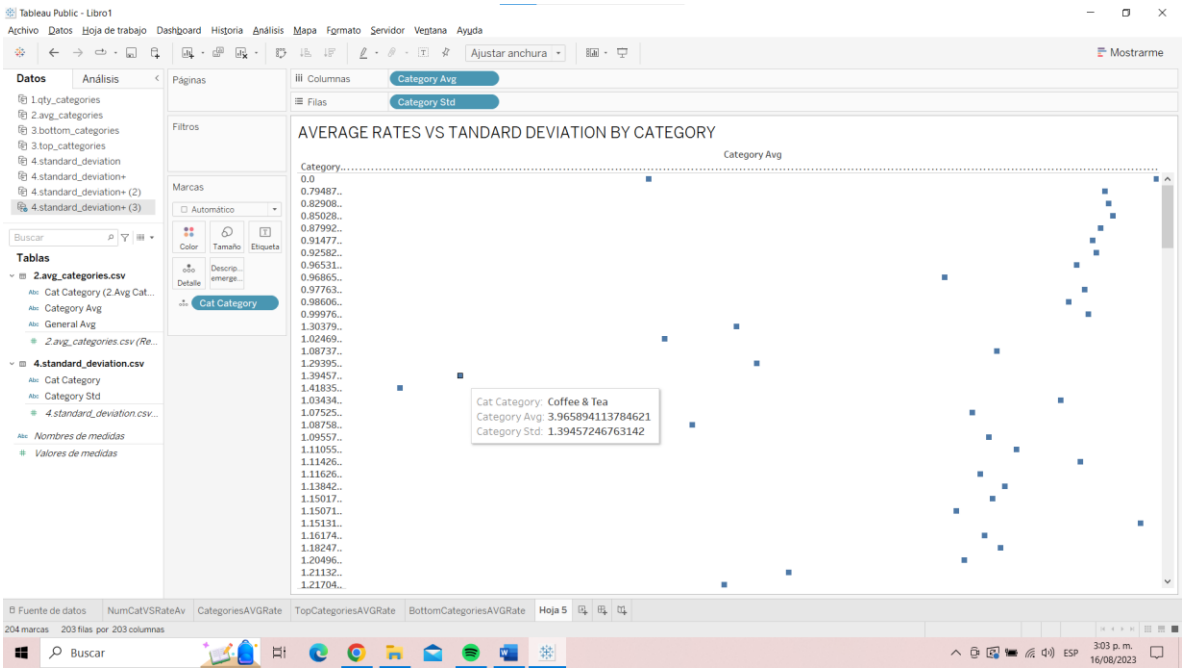
An apparent trend was found that the categories with the highest average have less deviation, and the categories with the lowest average have greater deviation; however, the presence of enough data that do not follow this pattern indicates that the relationship is not necessarily absolute and that there is other factors to consider, for example:

Diversity of Experiences: Categories that are not following the trend may be experiencing a wide range of experiences and opinions. This could indicate that customer experiences within these categories are diverse and that there are factors that drive widely varying opinions.

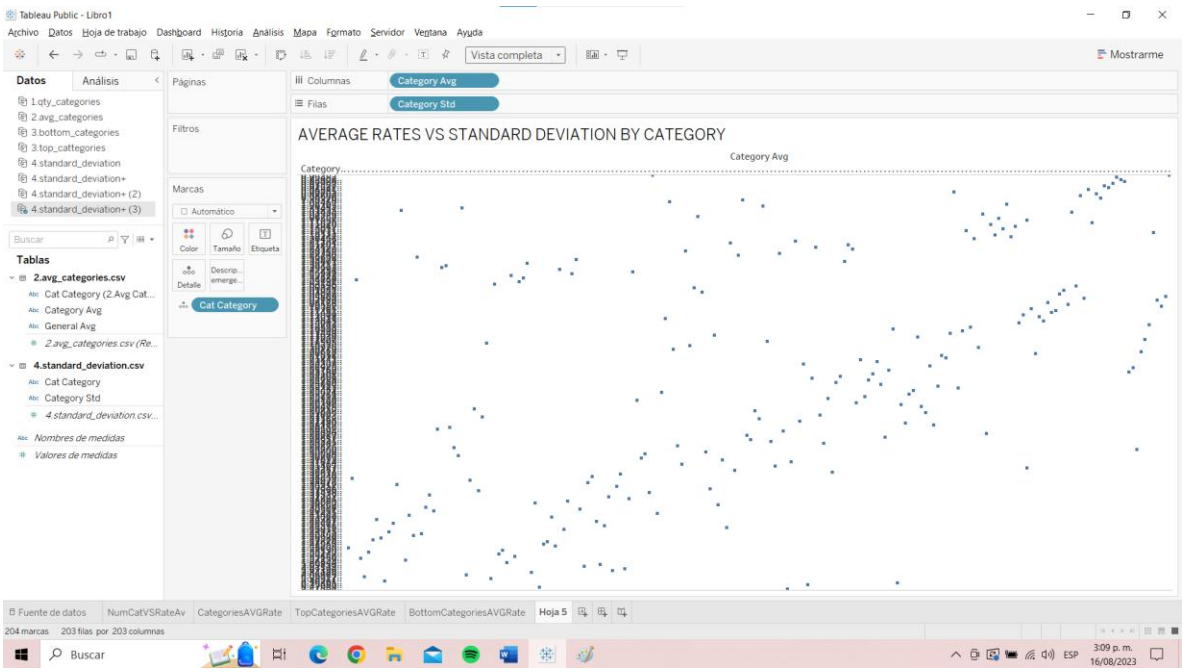
Influencing Factors: Non-trending data may be influenced by unique factors that do not apply to trending categories. There may be particular characteristics of those categories that affect the relationship between the mean and the deviation.

In conclusion, the relationship between the average rate and the standard deviation is complex and cannot be uniformly generalized to all categories. It is essential to take into account the specific context of each category to understand why some follow the trend and others do not.

Average Rates VS. Standard Deviation by Category



A little Zoom Out :



Map Reduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets. It is designed to scale up from single computers to thousands of machines. MapReduce programs are written in Java, Python, or other languages that can be compiled to Java bytecode.

The MapReduce programming model consists of two parts:

- The map function: This function takes a key/value pair as input and produces a set of intermediate key/value pairs as output.
- The reduce function: This function takes a set of intermediate key/value pairs as input and produces a final set of key/value pairs as output.

The benefits of MapReduce include:

Scalability: MapReduce can be scaled up to process very large data sets.

Fault tolerance: MapReduce is fault-tolerant, meaning that it can continue to process data even

Here are some of the use cases of MapReduce:

Analyzing large data sets, such as financial transactions, customer logs, or social media data.

Distributing tasks across a cluster of computers.

Processing data in real time.

Generating reports and summaries.

Storing and retrieving data.

Due this ability of Map Reduce we are going to run some jobs using that to be more familiar with the capabilities

Running some jobs using Map Reduce

A Map Reduce job ran to count the words on top-rated reviews and low created reviews

Counting the words in top-rated reviews and low-rated reviews can be helpful for data analysts in a number of ways. Here are a few examples:

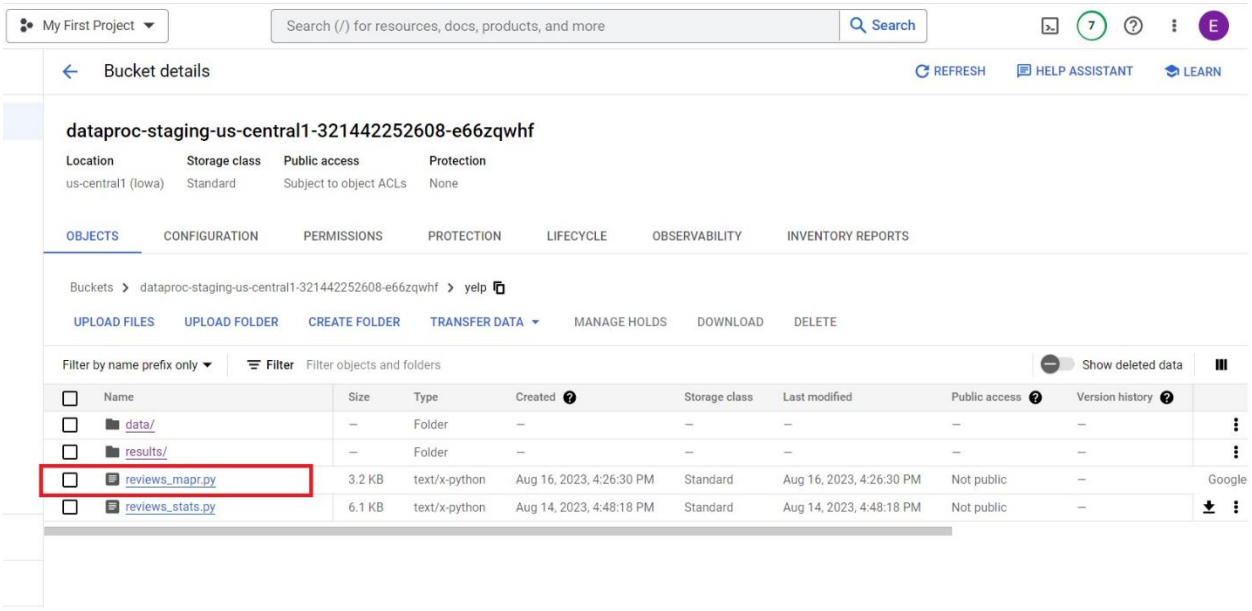
Identifying the most common words in each category. This can help analysts to understand the key themes and topics that are discussed in the reviews. For example, if the most common words in top-rated reviews are "food," "service," and "ambiance," then analysts can infer that these are the factors that are most important to customers when rating a restaurant.

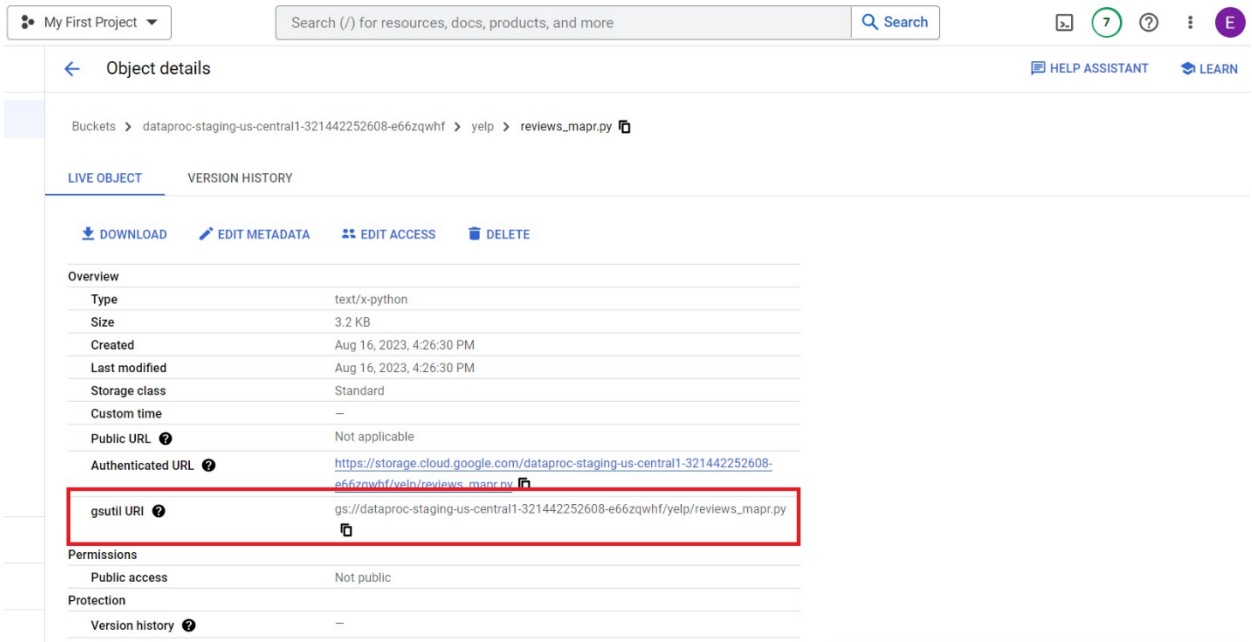
Comparing the word counts between the two categories. This can help analysts to identify any differences in the language used to describe top-rated and low-rated reviews. For example, if the most common words in low-rated reviews are "disappointing" and "unfriendly," then analysts can infer that these are the factors that are most likely to lead to a low rating.

Identifying outliers. If there are any words that are much more common in one category than the other, then these could be considered outliers. Outliers can be helpful for identifying unusual patterns or trends in the data.

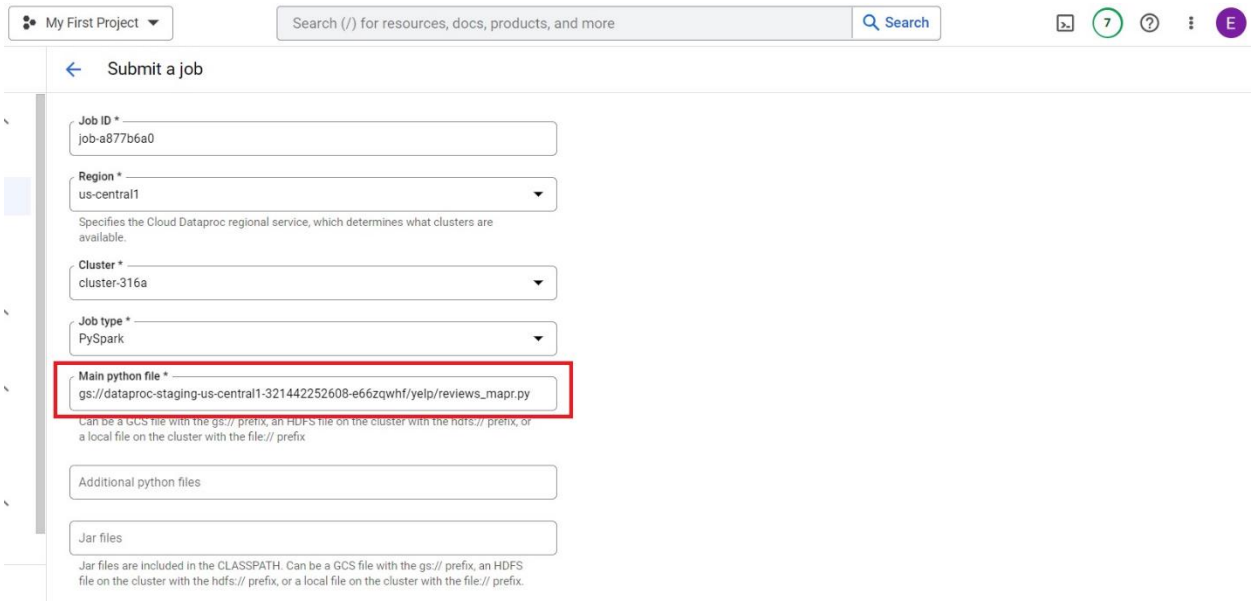
Generating reports and visualizations. The word counts can be used to generate reports and visualizations that can be used to communicate the findings of the analysis to stakeholders.

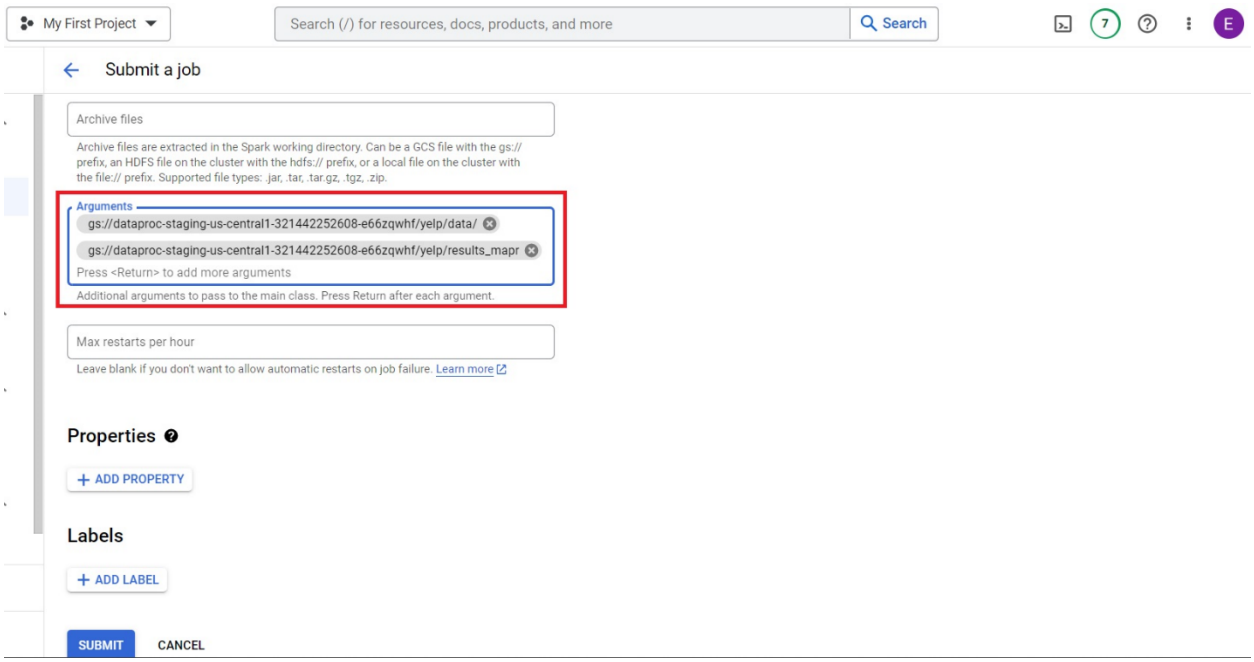
Uploading a new python program to getting reviews from categories which are over and under the general avg rate. This will be the input for map reduce process.



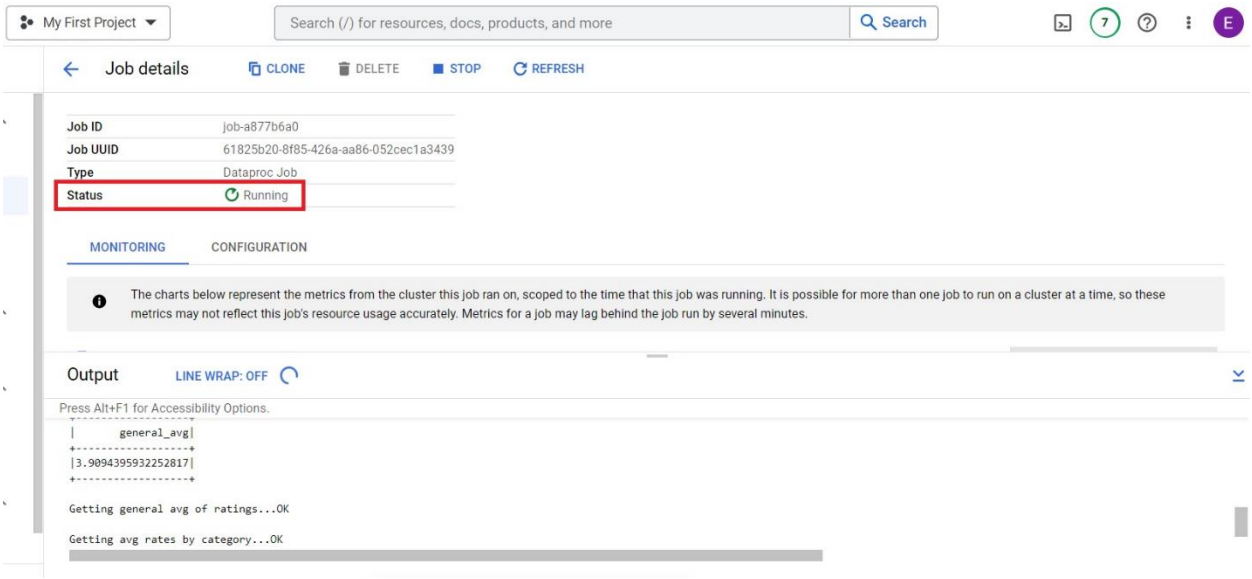


Submitting a new job

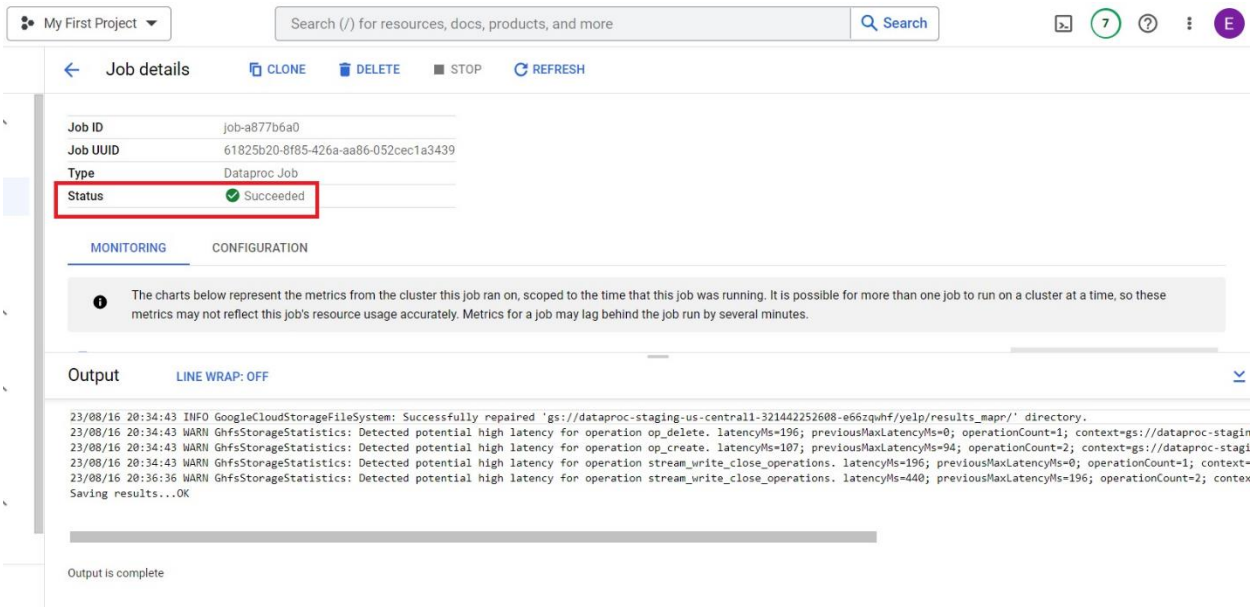




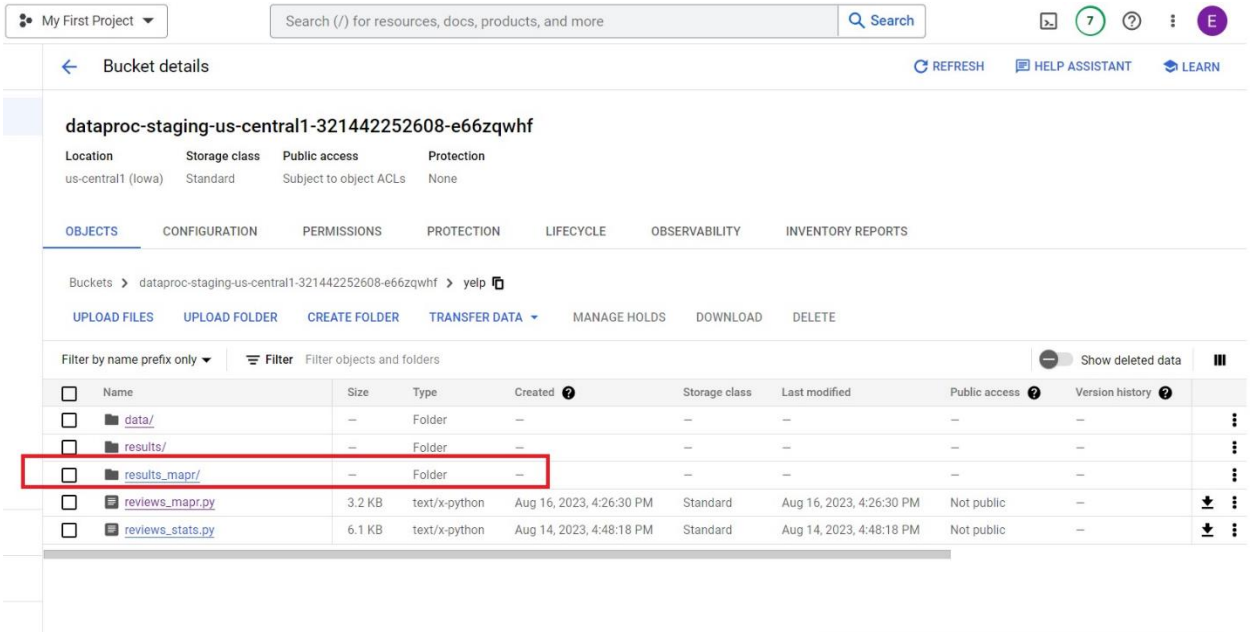
Running the job



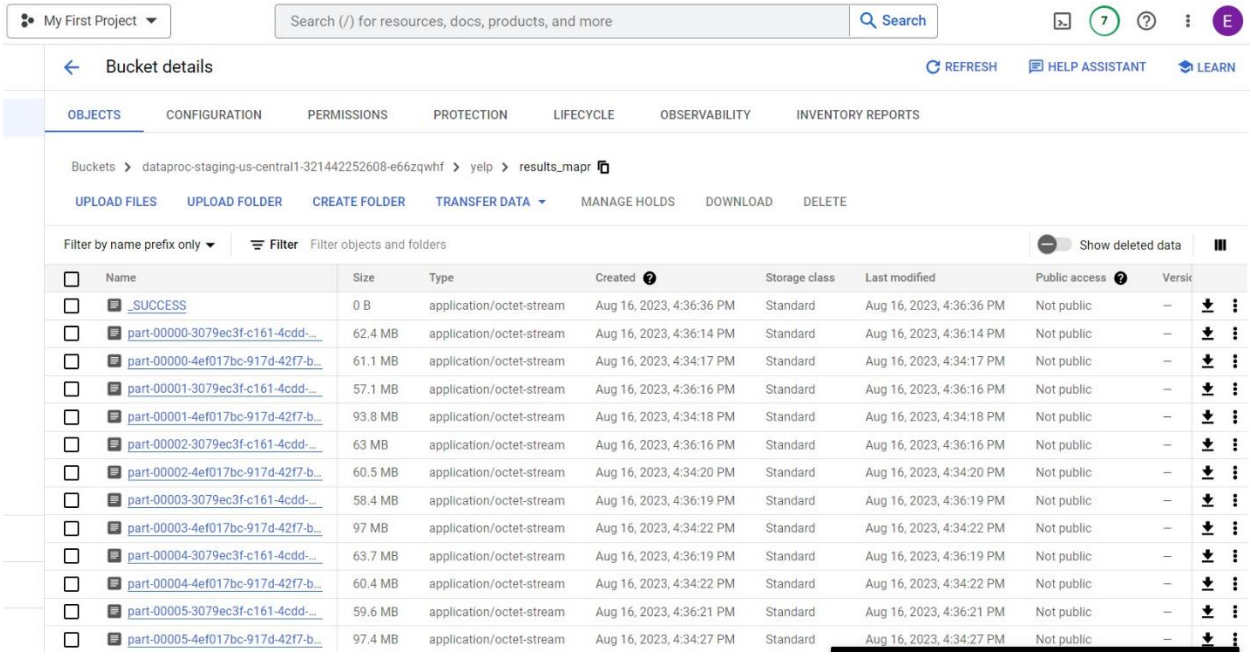
Job successfully completed



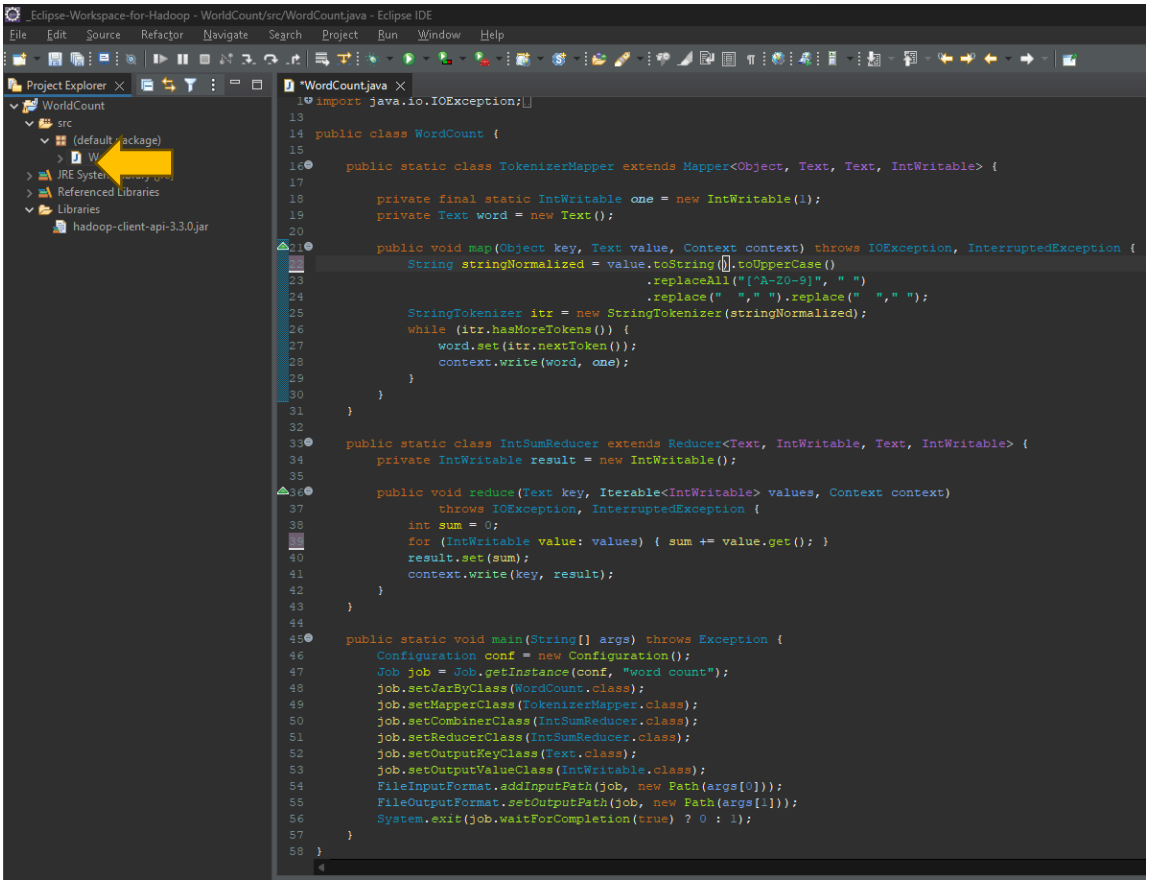
Results folder



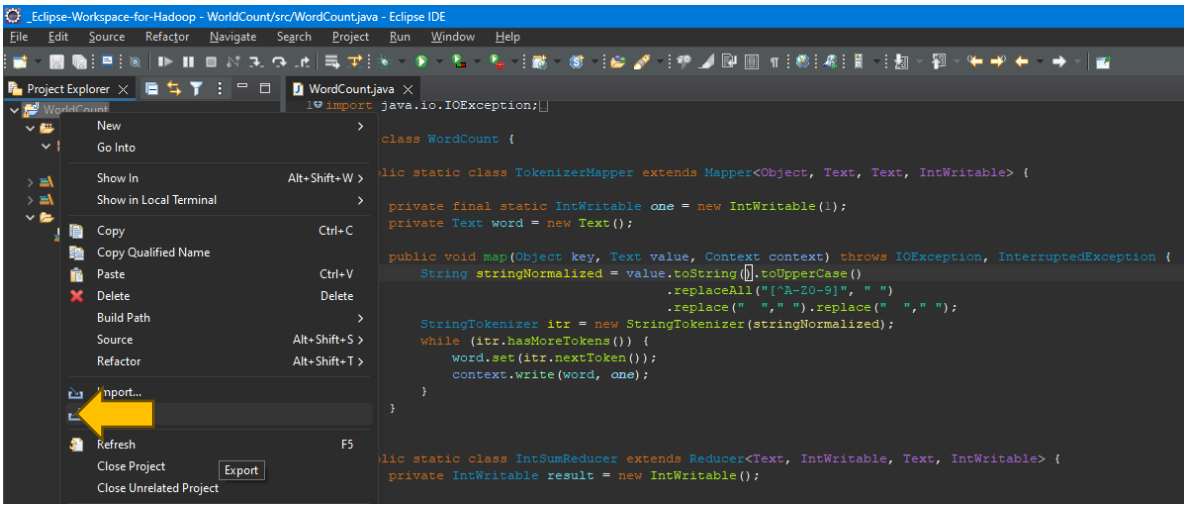
Resulting csv files



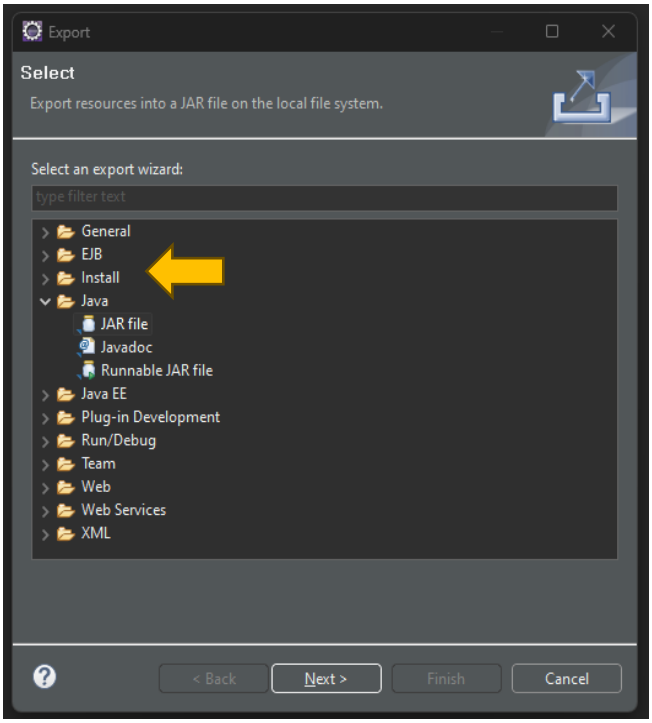
Step 01. Prepare the MapReduce Java Application. In this case using Eclipse IDE.



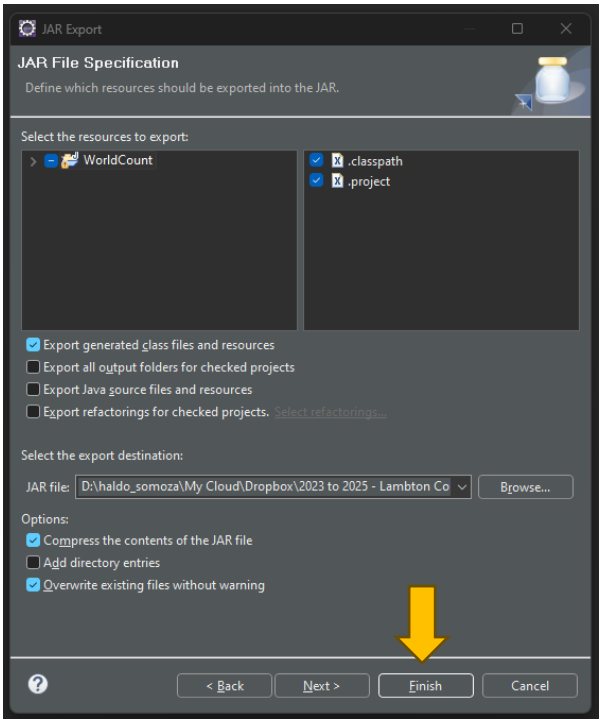
Right click over the project, choose Export



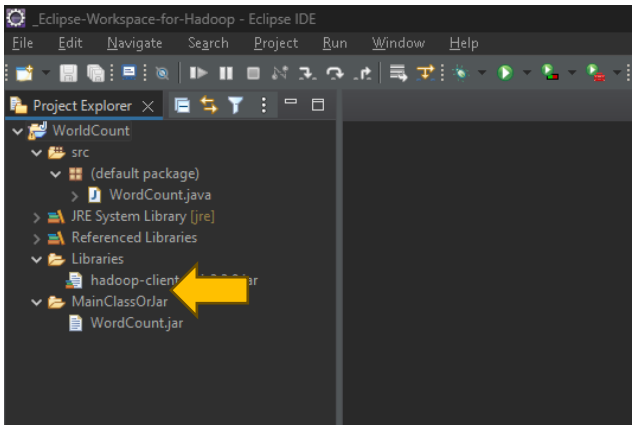
Choose Java -> JAR file. Press Next button.



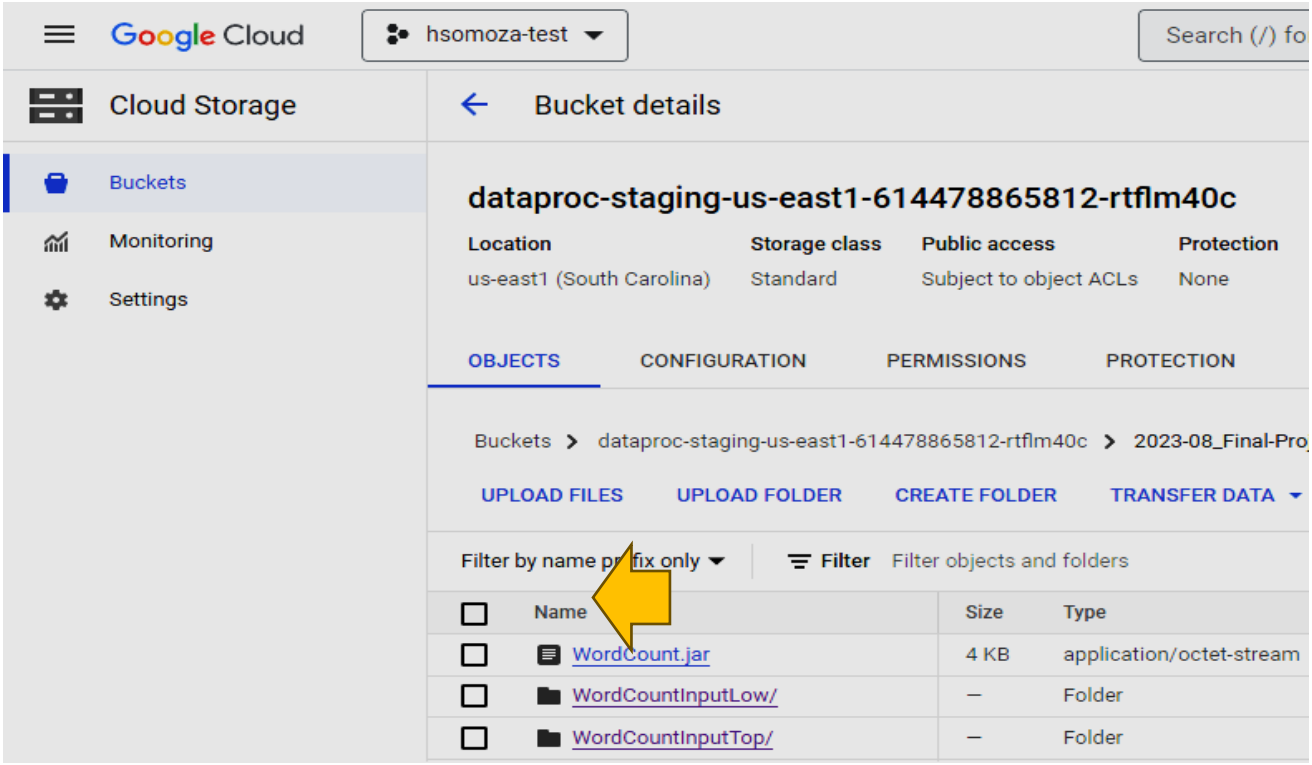
Select options like you see in the image and define one export destination folder and filename for the resulting JAR file. Then click on Finish.



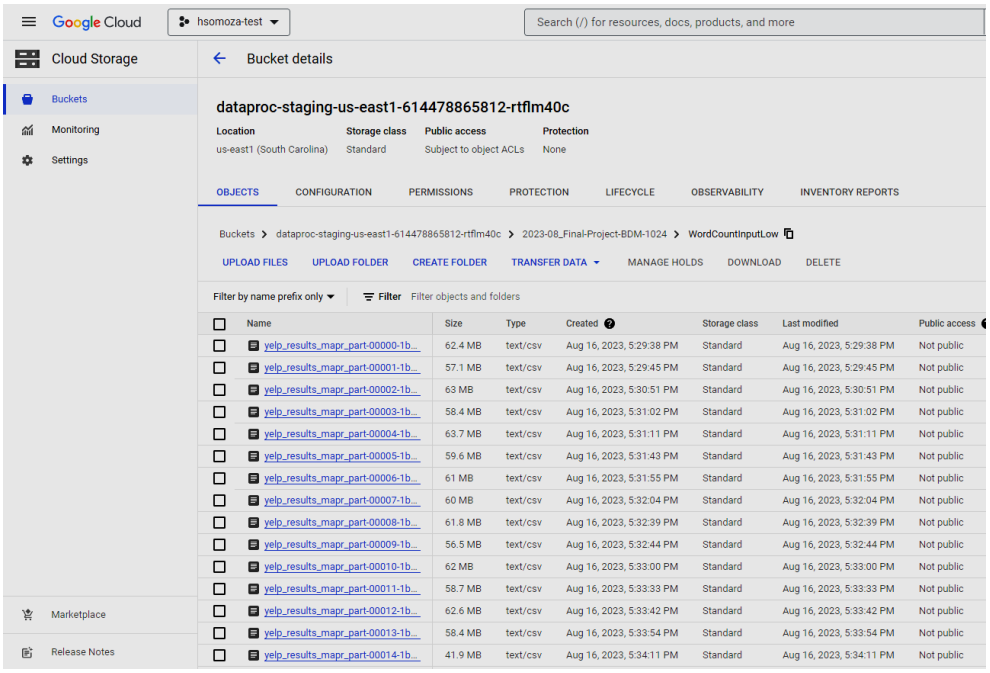
Verify the generated JAR file.



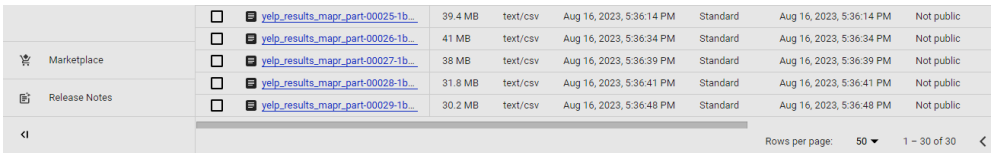
Step 02: Upload the JAR file and the Input Data folders to a GCP Bucket.



In this case upload two Input Data folders, one to process data with low ratings and another to process data with top ratings. In next image the WordCountInputLow folder with data generated by PySpark.

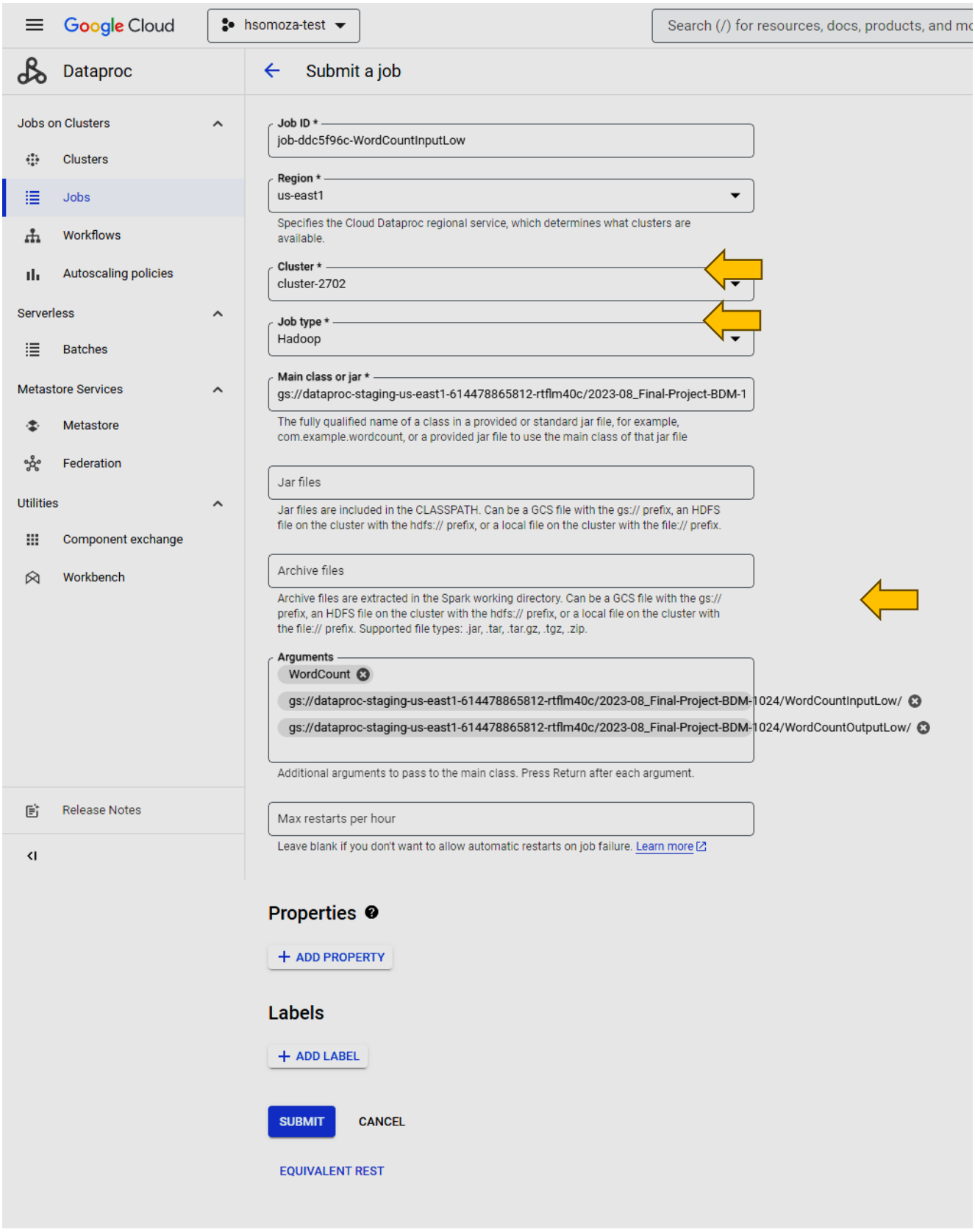


...



Step 03: Creating and Executing one Hadoop Job for each two folders.

In Job definition window, choose Hadoop Job type, write the path of jar in Main class or jar box, and such Arguments send (a) the Class Name, (b) the Input Data folder, and (c) the Output Data folder.


Google Cloud
hsomoza-test
Search (/) for resources, docs, products, and mc
Dataproc
Submit a job
Jobs on Clusters
Clusters
Jobs
Workflows
Autoscaling policies
Serverless
Batches
Metastore Services
Metastore
Federation
Utilities
Component exchange
Workbench
Release Notes
Job ID *
job-ddc5f96c-WordCountInputLow
Region *
us-east1
Specifies the Cloud Dataproc regional service, which determines what clusters are available.
Cluster *
cluster-2702
Job type *
Hadoop
Main class or jar *
gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1
The fully qualified name of a class in a provided or standard jar file, for example, com.example.wordcount, or a provided jar file to use the main class of that jar file
Jar files
Jar files are included in the CLASSPATH. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix.
Archive files
Archive files are extracted in the Spark working directory. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.
Arguments
WordCount
gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountInputLow/
gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountOutputLow/
Additional arguments to pass to the main class. Press Return after each argument.
Max restarts per hour
Leave blank if you don't want to allow automatic restarts on job failure. Learn more
Properties
ADD PROPERTY
Labels
ADD LABEL
SUBMIT
CANCEL
EQUIVALENT REST

The jobs finished successfully, showing the following information and log, similarly in both jobs.

The log resulting:

2023-08-16 22:29:28,765 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-2702-m.us-east1-b.c.hsomoza-test.internal./10.142.0.4:8032

2023-08-16 22:29:29,066 INFO client.AHSProxy: Connecting to Application History server at cluster-2702-m.us-east1-b.c.hsomoza-test.internal./10.142.0.4:10200

Aug 16, 2023 10:29:29 PM com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics updateStats

WARNING: Detected potential high latency for operation op_get_file_status. latencyMs=361; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountOutputLow

2023-08-16 22:29:30,185 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.

2023-08-16 22:29:30,205 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1692216534885_0007

2023-08-16 22:29:30,684 INFO input.FileInputFormat: Total input files to process : 30

2023-08-16 22:29:30,795 INFO mapreduce.JobSubmitter: number of splits:30

2023-08-16 22:29:31,072 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1692216534885_0007

2023-08-16 22:29:31,072 INFO mapreduce.JobSubmitter: Executing with tokens: []

2023-08-16 22:29:31,317 INFO conf.Configuration: resource-types.xml not found

2023-08-16 22:29:31,317 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.

2023-08-16 22:29:31,597 INFO impl.YarnClientImpl: Submitted application application_1692216534885_0007

2023-08-16 22:29:31,630 INFO mapreduce.Job: The url to track the job: http://cluster-2702-m.us-east1-b.c.hsomoza-test.internal.:8088/proxy/application_1692216534885_0007/

2023-08-16 22:29:31,631 INFO mapreduce.Job: Running job: job_1692216534885_0007

2023-08-16 22:29:42,740 INFO mapreduce.Job: Job job_1692216534885_0007 running in uber mode : false

2023-08-16 22:29:42,741 INFO mapreduce.Job: map 0% reduce 0%

2023-08-16 22:30:00,899 INFO mapreduce.Job: map 2% reduce 0%

2023-08-16 22:30:05,926 INFO mapreduce.Job: map 4% reduce 0%

2023-08-16 22:30:10,977 INFO mapreduce.Job: map 5% reduce 0%

2023-08-16 22:30:11,983 INFO mapreduce.Job: map 7% reduce 0%

```
2023-08-16 22:30:29,082 INFO mapreduce.Job:  map 9% reduce 0%
2023-08-16 22:30:30,087 INFO mapreduce.Job:  map 10% reduce 0%
2023-08-16 22:30:33,102 INFO mapreduce.Job:  map 11% reduce 0%
2023-08-16 22:30:34,115 INFO mapreduce.Job:  map 12% reduce 0%
2023-08-16 22:30:39,145 INFO mapreduce.Job:  map 13% reduce 0%
2023-08-16 22:30:54,226 INFO mapreduce.Job:  map 14% reduce 0%
2023-08-16 22:30:56,234 INFO mapreduce.Job:  map 16% reduce 0%
2023-08-16 22:30:57,239 INFO mapreduce.Job:  map 17% reduce 0%
2023-08-16 22:31:00,255 INFO mapreduce.Job:  map 18% reduce 0%
2023-08-16 22:31:06,284 INFO mapreduce.Job:  map 20% reduce 0%
2023-08-16 22:31:18,350 INFO mapreduce.Job:  map 21% reduce 0%
2023-08-16 22:31:23,377 INFO mapreduce.Job:  map 24% reduce 0%
2023-08-16 22:31:24,382 INFO mapreduce.Job:  map 25% reduce 0%
2023-08-16 22:31:32,420 INFO mapreduce.Job:  map 27% reduce 0%
2023-08-16 22:31:44,483 INFO mapreduce.Job:  map 28% reduce 0%
2023-08-16 22:31:46,492 INFO mapreduce.Job:  map 29% reduce 0%
2023-08-16 22:31:50,510 INFO mapreduce.Job:  map 31% reduce 0%
2023-08-16 22:31:52,517 INFO mapreduce.Job:  map 32% reduce 0%
2023-08-16 22:31:58,538 INFO mapreduce.Job:  map 33% reduce 0%
2023-08-16 22:31:59,542 INFO mapreduce.Job:  map 34% reduce 0%
2023-08-16 22:32:11,599 INFO mapreduce.Job:  map 36% reduce 0%
2023-08-16 22:32:12,605 INFO mapreduce.Job:  map 37% reduce 0%
2023-08-16 22:32:17,627 INFO mapreduce.Job:  map 39% reduce 0%
2023-08-16 22:32:25,658 INFO mapreduce.Job:  map 40% reduce 0%
2023-08-16 22:32:31,681 INFO mapreduce.Job:  map 41% reduce 0%
2023-08-16 22:32:33,689 INFO mapreduce.Job:  map 42% reduce 0%
2023-08-16 22:32:37,705 INFO mapreduce.Job:  map 43% reduce 0%
2023-08-16 22:32:43,730 INFO mapreduce.Job:  map 46% reduce 0%
```

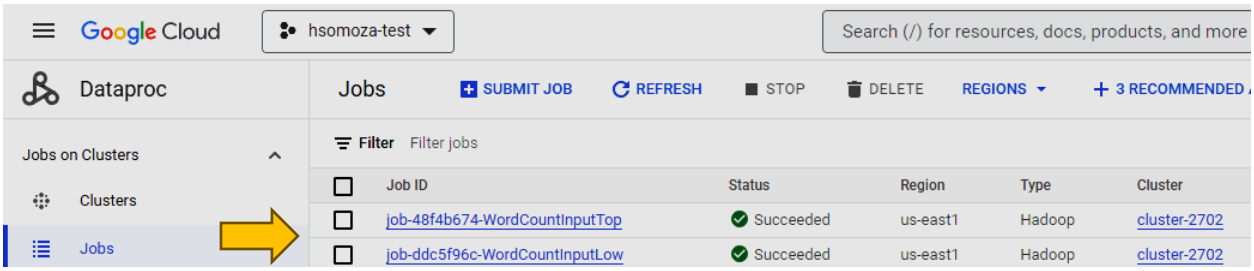
```
2023-08-16 22:32:48,750 INFO mapreduce.Job:  map 47% reduce 0%
2023-08-16 22:32:55,780 INFO mapreduce.Job:  map 48% reduce 0%
2023-08-16 22:32:58,796 INFO mapreduce.Job:  map 49% reduce 0%
2023-08-16 22:32:59,799 INFO mapreduce.Job:  map 50% reduce 0%
2023-08-16 22:33:06,826 INFO mapreduce.Job:  map 52% reduce 0%
2023-08-16 22:33:10,840 INFO mapreduce.Job:  map 53% reduce 0%
2023-08-16 22:33:18,874 INFO mapreduce.Job:  map 55% reduce 0%
2023-08-16 22:33:20,880 INFO mapreduce.Job:  map 56% reduce 0%
2023-08-16 22:33:24,893 INFO mapreduce.Job:  map 57% reduce 0%
2023-08-16 22:33:26,901 INFO mapreduce.Job:  map 58% reduce 0%
2023-08-16 22:33:27,905 INFO mapreduce.Job:  map 60% reduce 0%
2023-08-16 22:33:30,917 INFO mapreduce.Job:  map 61% reduce 0%
2023-08-16 22:33:36,939 INFO mapreduce.Job:  map 62% reduce 0%
2023-08-16 22:33:37,942 INFO mapreduce.Job:  map 63% reduce 0%
2023-08-16 22:33:47,974 INFO mapreduce.Job:  map 66% reduce 0%
2023-08-16 22:33:50,984 INFO mapreduce.Job:  map 67% reduce 0%
2023-08-16 22:33:57,002 INFO mapreduce.Job:  map 68% reduce 0%
2023-08-16 22:34:00,018 INFO mapreduce.Job:  map 69% reduce 0%
2023-08-16 22:34:03,028 INFO mapreduce.Job:  map 70% reduce 0%
2023-08-16 22:34:06,037 INFO mapreduce.Job:  map 71% reduce 0%
2023-08-16 22:34:08,043 INFO mapreduce.Job:  map 73% reduce 0%
2023-08-16 22:34:11,055 INFO mapreduce.Job:  map 74% reduce 0%
2023-08-16 22:34:15,072 INFO mapreduce.Job:  map 76% reduce 0%
2023-08-16 22:34:16,076 INFO mapreduce.Job:  map 77% reduce 0%
2023-08-16 22:34:29,120 INFO mapreduce.Job:  map 79% reduce 0%
2023-08-16 22:34:32,129 INFO mapreduce.Job:  map 80% reduce 0%
2023-08-16 22:34:36,141 INFO mapreduce.Job:  map 81% reduce 0%
2023-08-16 22:34:38,147 INFO mapreduce.Job:  map 83% reduce 0%
```

```
2023-08-16 22:34:42,160 INFO mapreduce.Job:  map 84% reduce 0%
2023-08-16 22:34:49,180 INFO mapreduce.Job:  map 87% reduce 0%
2023-08-16 22:34:51,188 INFO mapreduce.Job:  map 88% reduce 0%
2023-08-16 22:34:53,196 INFO mapreduce.Job:  map 90% reduce 0%
2023-08-16 22:35:08,254 INFO mapreduce.Job:  map 92% reduce 0%
2023-08-16 22:35:10,260 INFO mapreduce.Job:  map 93% reduce 0%
2023-08-16 22:35:15,274 INFO mapreduce.Job:  map 97% reduce 0%
2023-08-16 22:35:21,291 INFO mapreduce.Job:  map 98% reduce 0%
2023-08-16 22:35:26,305 INFO mapreduce.Job:  map 100% reduce 0%
2023-08-16 22:35:39,349 INFO mapreduce.Job:  map 100% reduce 33%
2023-08-16 22:35:45,366 INFO mapreduce.Job:  map 100% reduce 67%
2023-08-16 22:35:46,369 INFO mapreduce.Job:  map 100% reduce 100%
2023-08-16 22:35:47,377 INFO mapreduce.Job: Job job_1692216534885_0007 completed successfully
2023-08-16 22:35:47,506 INFO mapreduce.Job: Counters: 61
        File System Counters
                FILE: Number of bytes read=38735724
                FILE: Number of bytes written=71646869
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                GS: Number of bytes read=1538677788
                GS: Number of bytes written=2298801
                GS: Number of read operations=375670
                GS: Number of large read operations=0
                GS: Number of write operations=835544
                HDFS: Number of bytes read=7260
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=30
```

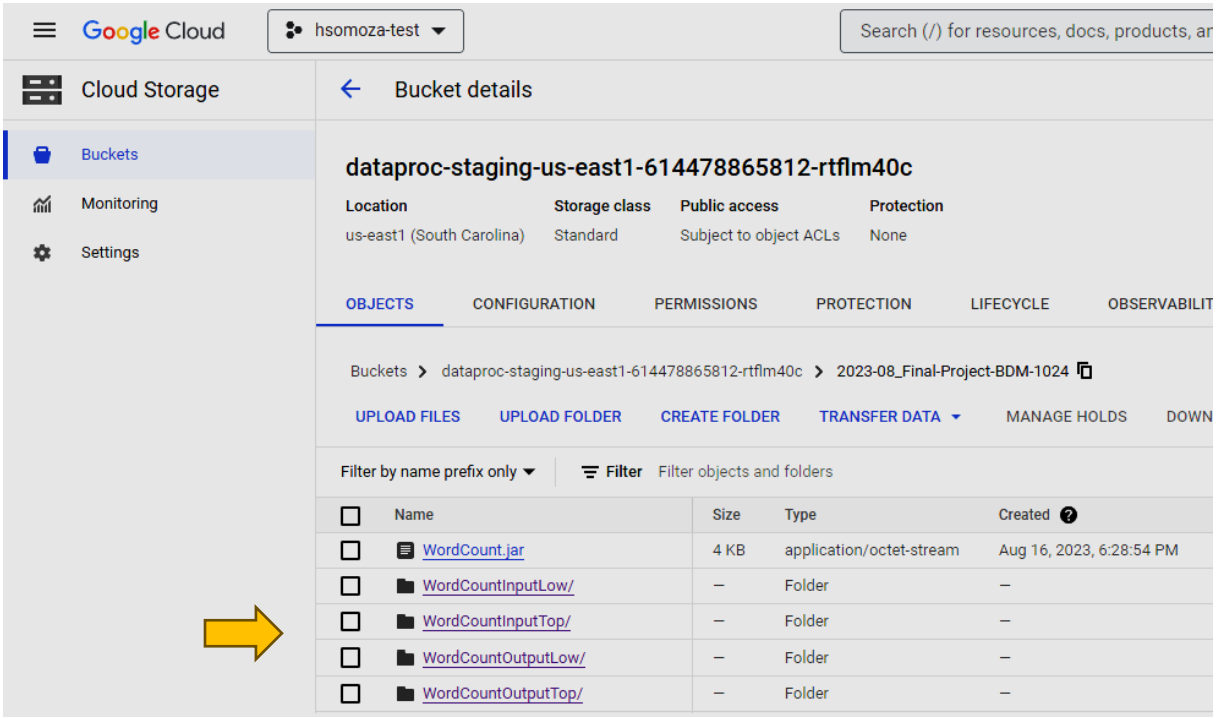
```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=1
                Killed reduce tasks=1
                Launched map tasks=30
                Launched reduce tasks=4
                Rack-local map tasks=30
                Total time spent by all maps in occupied slots (ms)=3221969339
                Total time spent by all reduces in occupied slots (ms)=155028316
                Total time spent by all map tasks (ms)=983207
                Total time spent by all reduce tasks (ms)=47308
                Total vcore-milliseconds taken by all map tasks=983207
                Total vcore-milliseconds taken by all reduce tasks=47308
                Total megabyte-milliseconds taken by all map tasks=3221969339
                Total megabyte-milliseconds taken by all reduce tasks=155028316
        Map-Reduce Framework
                Map input records=4326178
                Map output records=288118351
                Map output bytes=2640412774
                Map output materialized bytes=23539632
                Input split bytes=7260
                Combine input records=288118351
                Combine output records=1681325
                Reduce input groups=208886
                Reduce shuffle bytes=23539632
                Reduce input records=1681325
```

```
                Reduce output records=208886
                Spilled Records=4440584
                Shuffled Maps =90
                Failed Shuffles=0
                Merged Map outputs=90
                GC time elapsed (ms)=11007
                CPU time spent (ms)=851400
                Physical memory (bytes) snapshot=23218028544
                Virtual memory (bytes) snapshot=156121165824
                Total committed heap usage (bytes)=18877513728
                Peak Map Physical memory (bytes)=793665536
                Peak Map Virtual memory (bytes)=4743344128
                Peak Reduce Physical memory (bytes)=502362112
                Peak Reduce Virtual memory (bytes)=4730691584
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1538677788
        File Output Format Counters
                Bytes Written=2298801
```

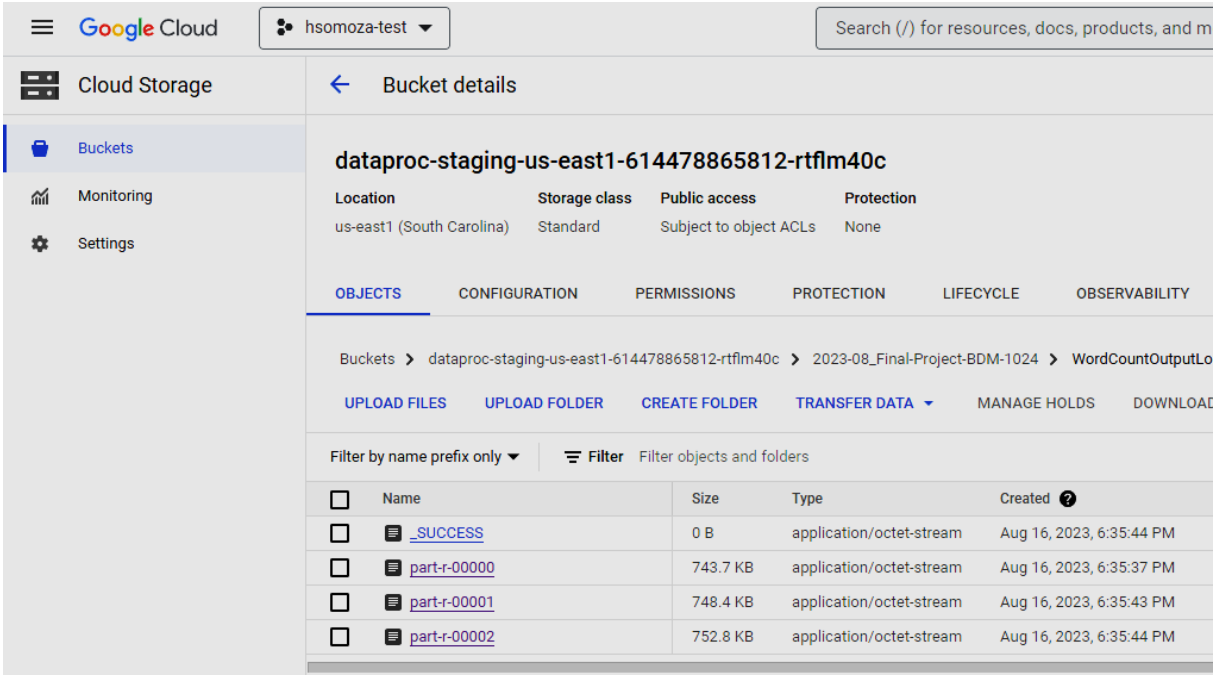
Both Hadoop jobs finished successfully.



Step 04: Identifying and Importing the resulting files.

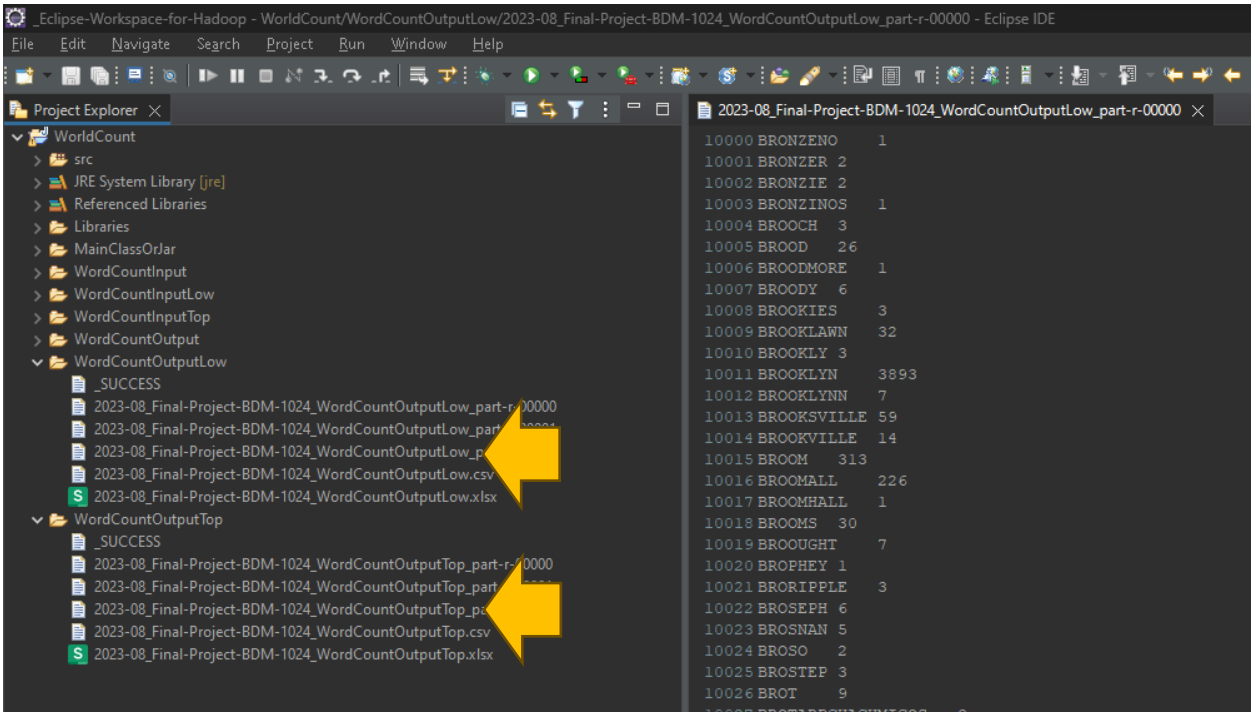


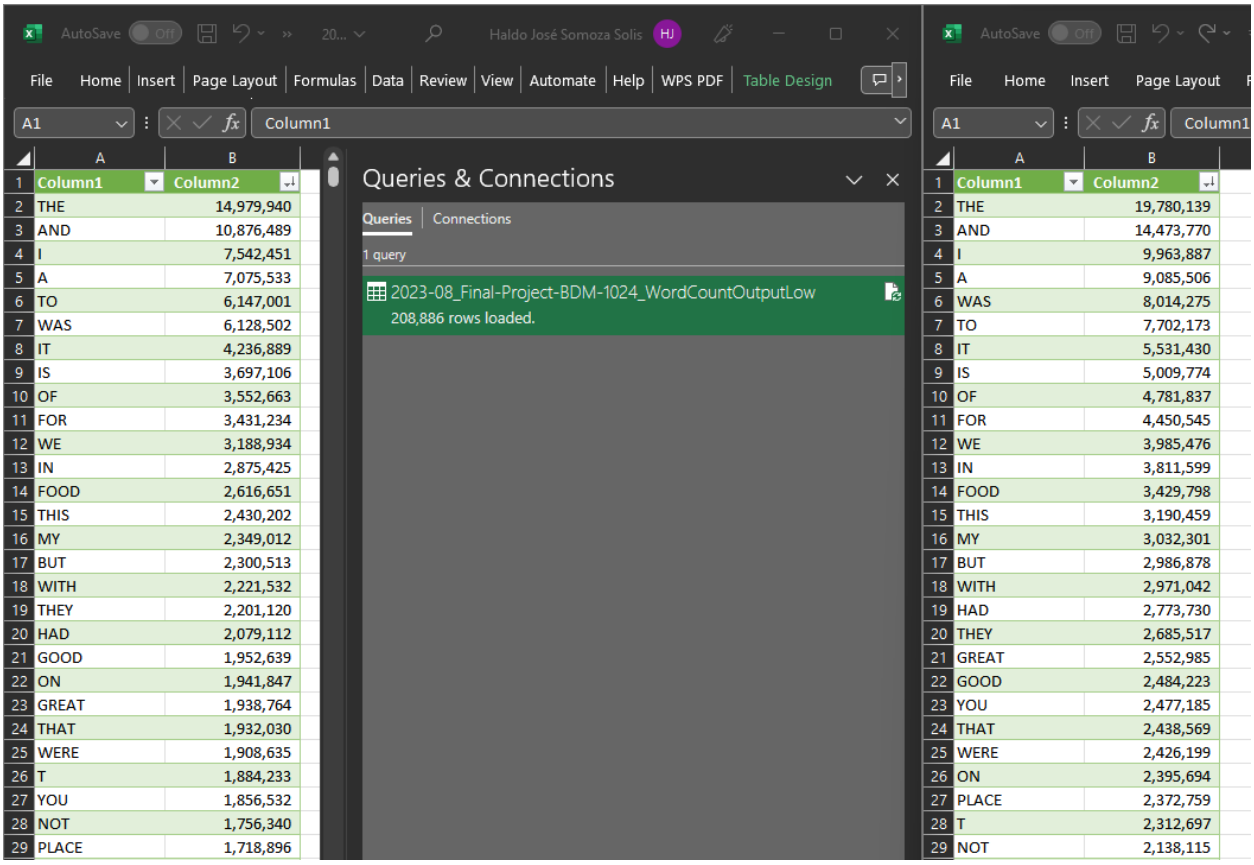
Here the result for one job splatted in three parts.



Step 05: Loading the data into one Excel file to analyze.

The downloaded data was joined in one only CSV file and then loaded into one Excel file.





	Column1	Column2
1	THE	14,979,940
2	AND	10,876,489
3	I	7,542,451
4	A	7,075,533
5	TO	6,147,001
6	WAS	6,128,502
7	IT	4,236,889
8	IS	3,697,106
9	OF	3,552,663
10	FOR	3,431,234
11	WE	3,188,934
12	IN	2,875,425
13	FOOD	2,616,651
14	THIS	2,430,202
15	MY	2,349,012
16	BUT	2,300,513
17	WITH	2,221,532
18	THEY	2,201,120
19	HAD	2,079,112
20	GOOD	1,952,639
21	ON	1,941,847
22	GREAT	1,938,764
23	THAT	1,932,030
24	WERE	1,908,635
25	T	1,884,233
26	YOU	1,856,532
27	NOT	1,756,340
28	PLACE	1,718,896

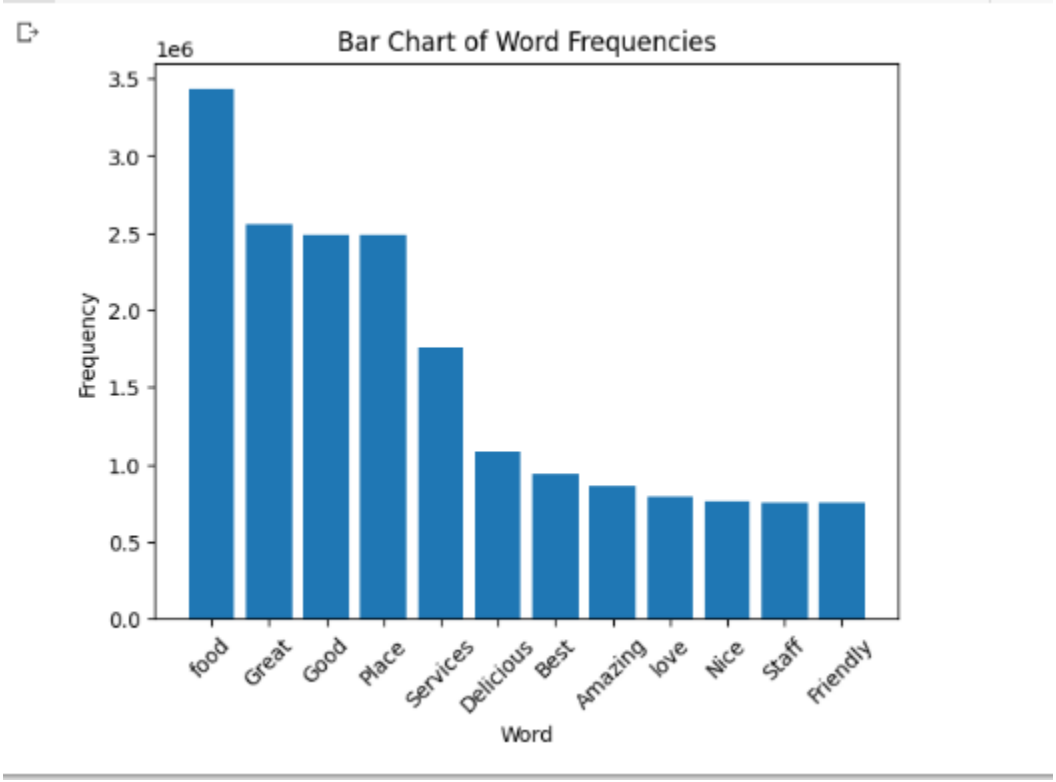
	Column1	Column2
1	THE	19,780,139
2	AND	14,473,770
3	I	9,963,887
4	A	9,085,506
5	WAS	8,014,275
6	TO	7,702,173
7	IT	5,531,430
8	IS	5,009,774
9	OF	4,781,837
10	FOR	4,450,545
11	WE	3,985,476
12	IN	3,811,599
13	FOOD	3,429,798
14	THIS	3,190,459
15	MY	3,032,301
16	BUT	2,986,878
17	WITH	2,971,042
18	HAD	2,773,730
19	THEY	2,685,517
20	GREAT	2,552,985
21	GOOD	2,484,223
22	YOU	2,477,185
23	THAT	2,438,569
24	WERE	2,426,199
25	ON	2,395,694
26	PLACE	2,372,759
27	T	2,312,697
28	NOT	2,138,115

By ran a MapReduce job to count the words of good_rated and bad_rated reviews we just understand that there are lots of common words that we need to clarify them by an advance Machine Learning algorithms to find out the specific result, so for now we just manually went trough all the result and remove the non-relative words that may not have a value in this result.

For example we found out for the good rated reviews people mostly used these worded

	A	B	C	D
1	Column1	Column2		
14	FOOD	3429798		
20	THEY	2685517		
21	GREAT	2552985		
22	GOOD	2484223		
27	PLACE	2372759		
35	SERVICE	1758807		
50	DELICIOUS	1087390		
56	BEST	943263		
61	AMAZING	857580		
69	LOVE	793568		
70	GOT	779726		
71	VE	768092		
72	NICE	767495		
73	BEEN	762117		
74	DEFINITELY	751551		
75	STAFF	751292		
76	FRIENDLY	751161		
77	CHICKEN	741094		
78	UP	736032		

Food, Place, Service, Staff …. So it just shows us people don't evaluate a restaurant just based on the food, they also pay attention to services the place and the manner of the staff as well.



The most common word in the bar graph is "food," with a frequency of 3429798. This suggests that food is the most important factor for customers when rating a restaurant. it also provide some insights into what customers are looking for in a restaurant. For example, the words "place," "services," and "staff" suggest that customers also value the atmosphere and service of a restaurant. The words "delicious," "best," and "amazing" suggest that customers are looking for a high-quality dining experience. Keep I mind that this is just top written words on good-rated reviews

Conclusions and Recommendations

It was identified that the city with the largest number of restaurants is Philadelphia with a total of 5,852 businesses in this category, with an approximate average of 3.56 stars. It was also found that 7 of the 10 cities with the largest number of restaurants have more than 2,000 business units of this type. It is recommended that the company make a strategic approach to increase the number of restaurants that want to pay for advertising in these cities, and likewise make a strategic plan to increase user participation in them. This benefits both financially and in growth and strengthening of the YELP brand.

1,553 restaurants were found with a high 5-star rating from their users, which represents about the 3% of the population. It is recommended that the company take advantage of this information to establish a closer collaboration with these featured restaurants. They could analyze what is making the difference in these restaurants and identify a strategy to help other business to success. Collaborations must be organized based on special and strategic marketing programs. Yelp will also benefit as recommending these highly rated restaurants on its website can increase user satisfaction and loyalty and strengthen its position as a trusted platform for finding the best restaurants.

56079 restaurant reviews were found, with a star rating equal to or less than 2. 35% of these reviews are related to the issues “Customer services” or “Clean” or “Wait”. MacDonald’s, Taco Bell, and Wendy’s are the most repetitive restaurants with rates under 2. It is recommended to develop strategic actions to correct or minimize problems in the 58, 45, and 296 restaurants rated with stars less than or

equal to 2 and more than 10 reviews on issues related to "customer service", "cleanliness" and "waits" respectively. This may include setting quality guidelines for Yelp-listed restaurants, providing additional training for customer service staff, and encouraging improved cleaning and customer service standards at partner establishments. The implementation of the recommendations at this point would mean achieving and strengthening the reputation of businesses on the platform and increasing customer satisfaction, which translates customer loyalty to the platform, reducing user turnover and lowering the costs associated with the acquisition of new users, which benefits the profitability of the company and could also lead to an increase in advertising exposure and therefore advertising revenue.

The 10 users with the largest number of fans were found, among these the one with the most followers has a total of 12497, approximately 29% more than the user with the second most fans. All these users have more than 2,000 followers, a significant number of fans. It is recommended to establish solid relationships with the most influential users and develop marketing plans that include creating advertising guidelines with added value for interested businesses, this would include recommending businesses based on valuable content, and interacting with other users, among others. This would allow YELP to charge special rates for this type of advertising due to the reach and reputation these influential users have, creating an additional revenue stream for the company.

The 100 users with the largest number of reviews were obtained, the first of which is called Fox and has 17,473 reviews. Yelp is encouraged to design different rewards programs for users who make a high number of reviews. This can include benefits such as exclusive discounts, special promotions, or even points that can be redeemed for additional prizes or incentives. They could also create referral programs,

among others. All this to improve loyalty to the platform, which can in turn generate an expansion of its user base and generate greater economic opportunities.

To conclude this analytical solution focus on customer rating enhancement that creates a mutually beneficial environment, improving Yelp's reputation and generating greater revenue opportunities for businesses and for Yelp as well.

Roles of Group Members

TASKS	RESPONSIBLES
Design the postgresql database (model diagram)	Eduardo Williams / Adriana Penaranda
Design a python program to transform the information from json to Postgresql database	Haldo Somoza / Nilesh Khurana
Create the database in the cloud	Haldo Somoza
Import the information to Postgresql in the cloud	Marzieh Mohammadi / Adriana Penaranda / Haldo Somza / Eduardo Williams
Cleaning and filtering. (Restaurants) and check the quality of the data	Ishika Sukhija / Carlos Rey
Design SQL queries	Carlos Rey / Luis Gutierrez / Ishika Sukhija
Making the Mid Point Presentation	Marzieh Mohammadi / Adriana Penaranda/ Ishika Sukhija
Spark Jobs	Eduardo Williams / Adriana Penaranda
Map Reduce	Haldo Somoza / Marizeh Mohammadi
Visualization	Carlos Rey / Ishika Sukhija / Luis Gutierrez
Final Report	Marizeh Mohammadi / Nilesh Khurana
Final Presentation	Adriana Penaranda

References

yelp. (2023). Retrieved from https://www.yelp.com/dataset