MSIN0166 Data Engineering Group Project

Group Number: 2

Word Count: 3000

Table of Contents

1.0 Introduction

This report aims to obtain new data that has not been gathered from multiple sources which include data mining from APIs and web scraping from websites such as Trip Advisor as well as deploying the infrastructure needed to deliver data for business analysis. This project aims to collect restaurant related information such as price, ratings, number of reviews, address etc. The aim is to create a dataset containing information about top restaurants in London. The information gathered will be useful to the general public to inform their dining choices, data scientists and students to develop predictive models or to analyse factors (such as location, cuisine type) that affect a restaurant's ratings for example. This information will also be insightful to potential business owners.

The data will then be processed and stored using appropriate data engineering storage techniques and will be inserted into Docker to ensure reproducibility. The project also includes the use of GitHub to ensure source version control. The advantages and limitations of the data engineering tools used will be highlighted as well as areas for further work. The link to the GitHub repository has been provided: https://github.com/uceis42/Data-Engineering-2022-Groupwork

The full ETL pipeline has been depicted below:

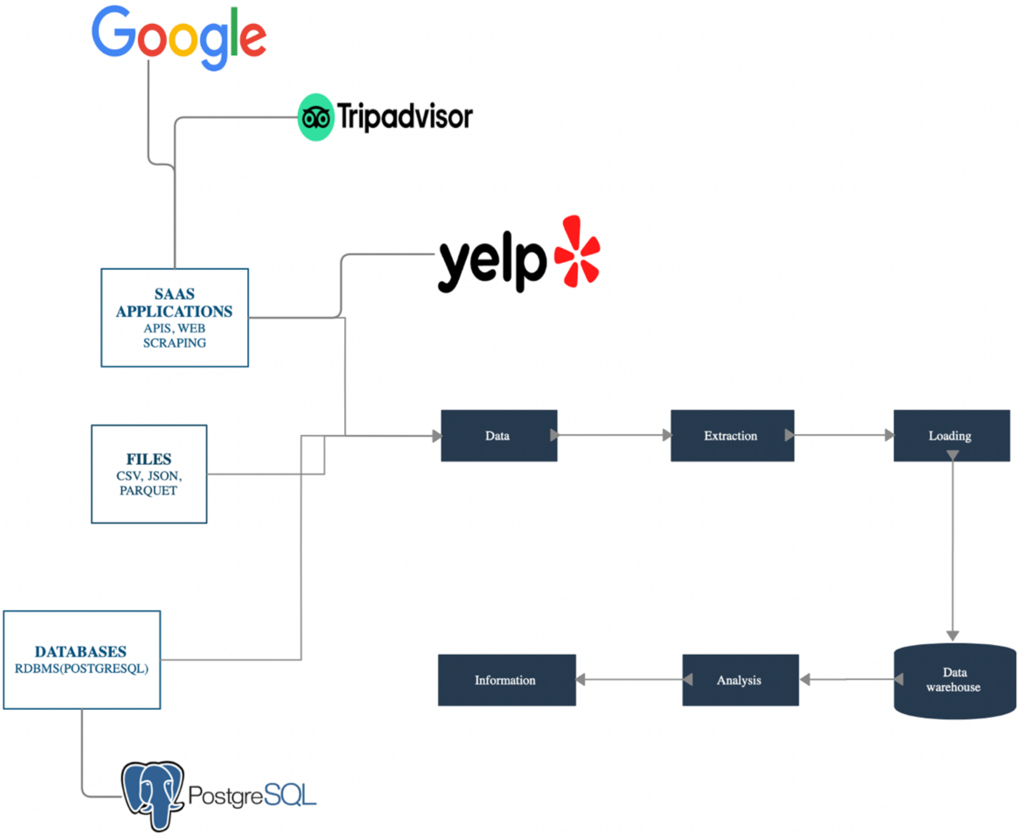


Figure 1. ETL Diagram

1.1 Data Gathering

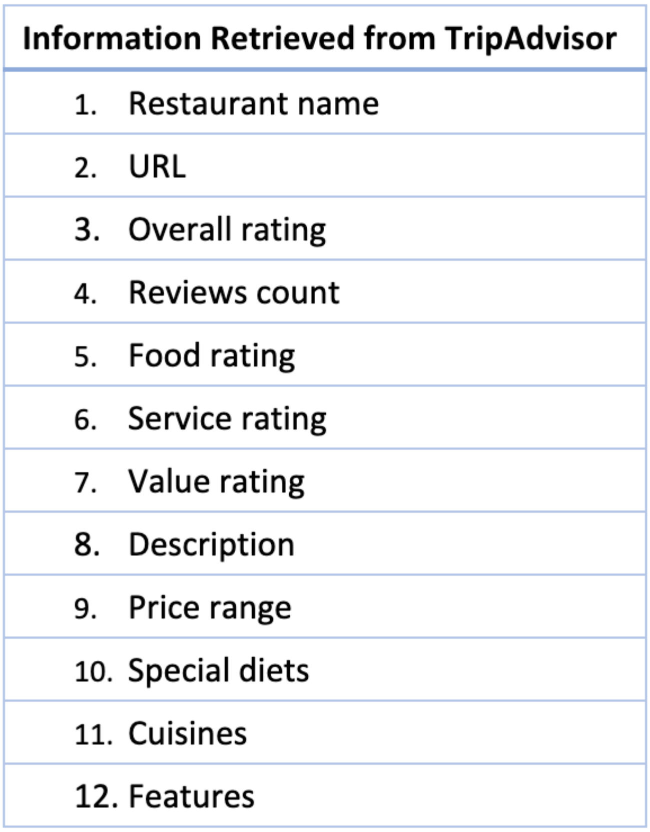
The data gathering process consisted of collecting data via restaurant centric API's and social media platforms such as Yelp, TripAdvisor, Google, and Food Standards Agency. APIs are highly efficient research tools and can aid with the collection, organisation, and cleaning of data (Lomborg and Bechmann, 2014). The data from TripAdvisor was extracted through web scraping.

1.1.1 TripAdvisor Data

Since TripAdvisor only provides APIs for non-commercial purpose, data are retrieved through web scraping. Python packages such as BeautifulSoup and requests are used in web scraping.

However, TripAdvisor has a strict policy against web scraping, thus there is only about 1/3 chance of establishing a successful connection with it even after trying different headers. At most 3 try and except are used to gain a higher chance of establishing a successful connection with TripAdvisor. Additionally, multiple web scraping results are combined. As a result, about 250/300 rows of data are successfully retrieved.

The limitation of using web scraping is that content on websites may not be arranged in the same way. In this case, some pages have less information than others. To get most of the information, a try and except method is implemented.



Information Retrieved from TripAdvisor
1. Restaurant name
2. URL
3. Overall rating
4. Reviews count
5. Food rating
6. Service rating
7. Value rating
8. Description
9. Price range
10. Special diets
11. Cuisines
12. Features

Table 1. TripAdvisor Table

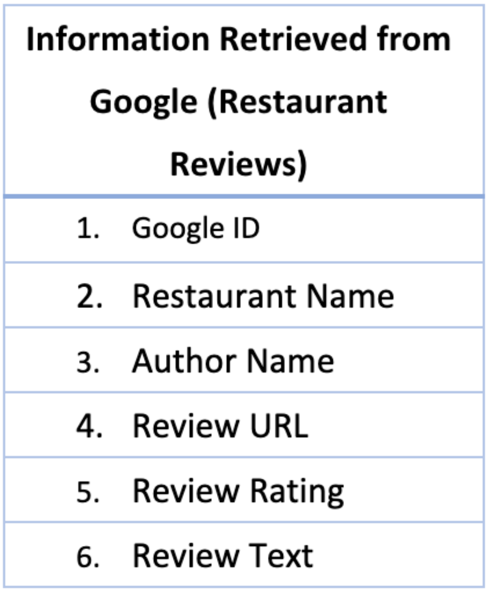
1.1.2 Google Data

The data retrieval process from Google's API required a two-step process and access to multiple endpoints for GET requests using REST API. The first API 'Google Place Search' only returns basic information such as address, name, and place id. It takes in a text string as input and returns the most relevant result. As only London restaurants were being searched 'London' was added to each restaurant name, to narrow the search. Additional fields/parameters can be requested, but it is still limited. Therefore, after the initial request, the retrieved Google place id is associated with a restaurant name. Subsequently, the Google place id is used as an input to fetch comprehensive establishment details from the second API ‘Google Place Details’ which includes complete address, phone number, website, user rating and reviews, see table 2 and 3. All this information is stored as a Json file.

In the database the Google data was stored as two separate tables, one with basic restaurant information and the other with the reviews. In the restaurant information table, the Google ID served as primary key. As for the Google reviews tables it had a serial self-incrementing tables as primary key that would assign an id to each review, but it would reference the Google restaurant information table using Google ID as foreign key, see figure 4 for the database schema. The separation was done to ensure that the database follows second normal form (2nf) of database normalisation, because having both restaurant information and reviews would lead to duplicated data which could lead to data redundancy.

Information Retrieved from Google (Restaurant Info)
1. Google ID
2. Restaurant Name
3. Address
4. Phone Number
5. Postcode
6. Price Level
7. Overall Rating
8. Total Ratings
9. Website URL

Table 2. Google Main Table.

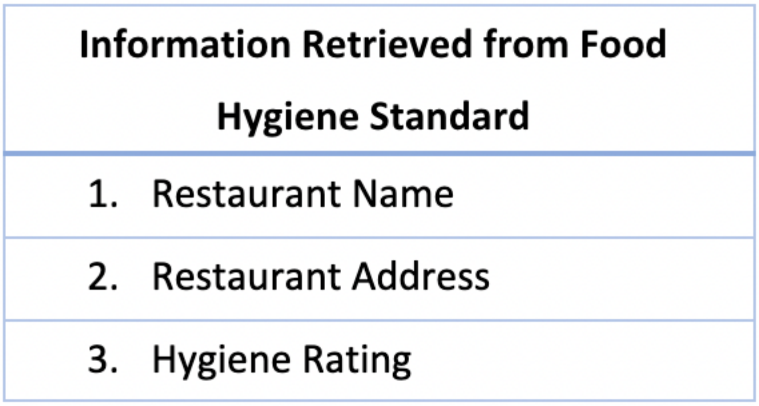


Information Retrieved from Google (Restaurant Reviews)
1. Google ID
2. Restaurant Name
3. Author Name
4. Review URL
5. Review Rating
6. Review Text

Table 3. Google Reviews Table

1.1.3 Food Standards Agency

The Food Standards Agency provides a free access to Food Hygiene Rating Scheme API (FHRS API) which allows retrieval of food hygiene rating data for all UK. The data is accessible through REST API with multiple endpoints and does not require any sign-up, API keys or logins. To ensure accurate GET requests for each restaurant, in addition to the restaurant name as input, the postcode obtained from the Google API was also passed into the search. The results were stored as Json files.



Information Retrieved from Food Hygiene Standard
1. Restaurant Name
2. Restaurant Address
3. Hygiene Rating

Table 4. Food Hygiene Table

1.1.4 Yelp Data.

To retrieve the required data from Yelp's API, GraphQL was used. GraphQL allows the retrieval of data using a single query to the server that includes concrete data requirements. By contrast, REST API gathers the required data using multiple endpoints. A major limitation of using REST API is the lack of flexibility as queries can only return fixed data structures which leads to either over-fetching (superfluous data) or under-fetching (minimal information).

Information Retrieved from Yelp
4. Yelp /Restaurant ID
5. Restaurant name
6. Review count
7. Rating
8. Price
9. Opening Hours

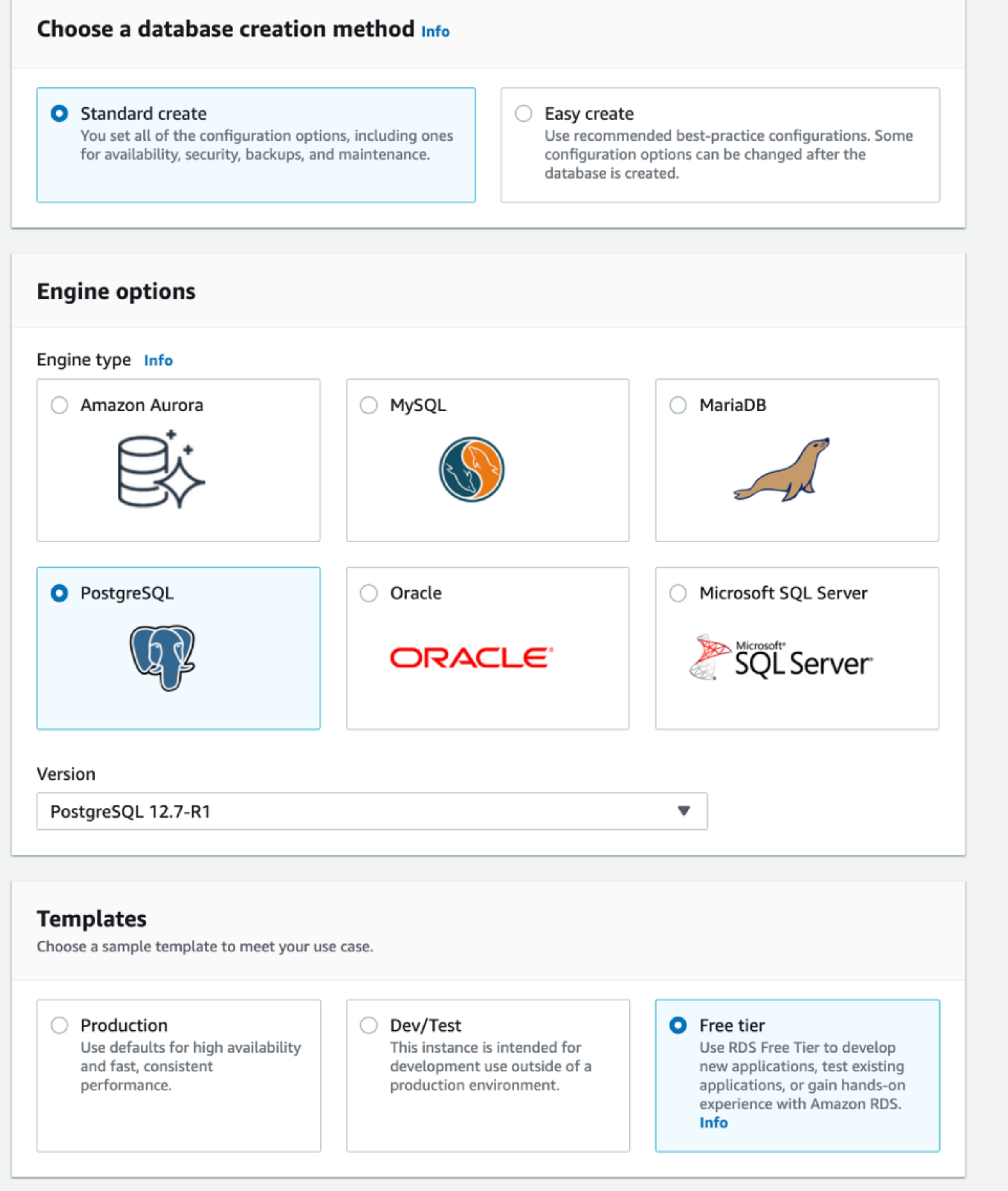
Table 5. Yelp Table

2.0 Data Storage and Processing

2.1 Data Storage and File Choice

After extracting the data from various APIS, the data was stored in csv and Json formats. However, after research on the industry's best practices—the data was stored as a Parquet file. Parquet files offer a significant benefit over both row-based and text-based file formats—such benefits include compression and optimised performance (Levy, 2022). This enables compression and encoding algorithms to take advantage of data type homogeneity to achieve better performance in terms of file size and speed (Ivanov and Pergolesi, 2020).

2.2 Database Selection



Choose a database creation method Info

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Amazon Aurora

○ MySQL

○ MariaDB

● PostgreSQL

○ Oracle

○ Microsoft SQL Server

Version

PostgreSQL 12.7-R1 ▼

Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

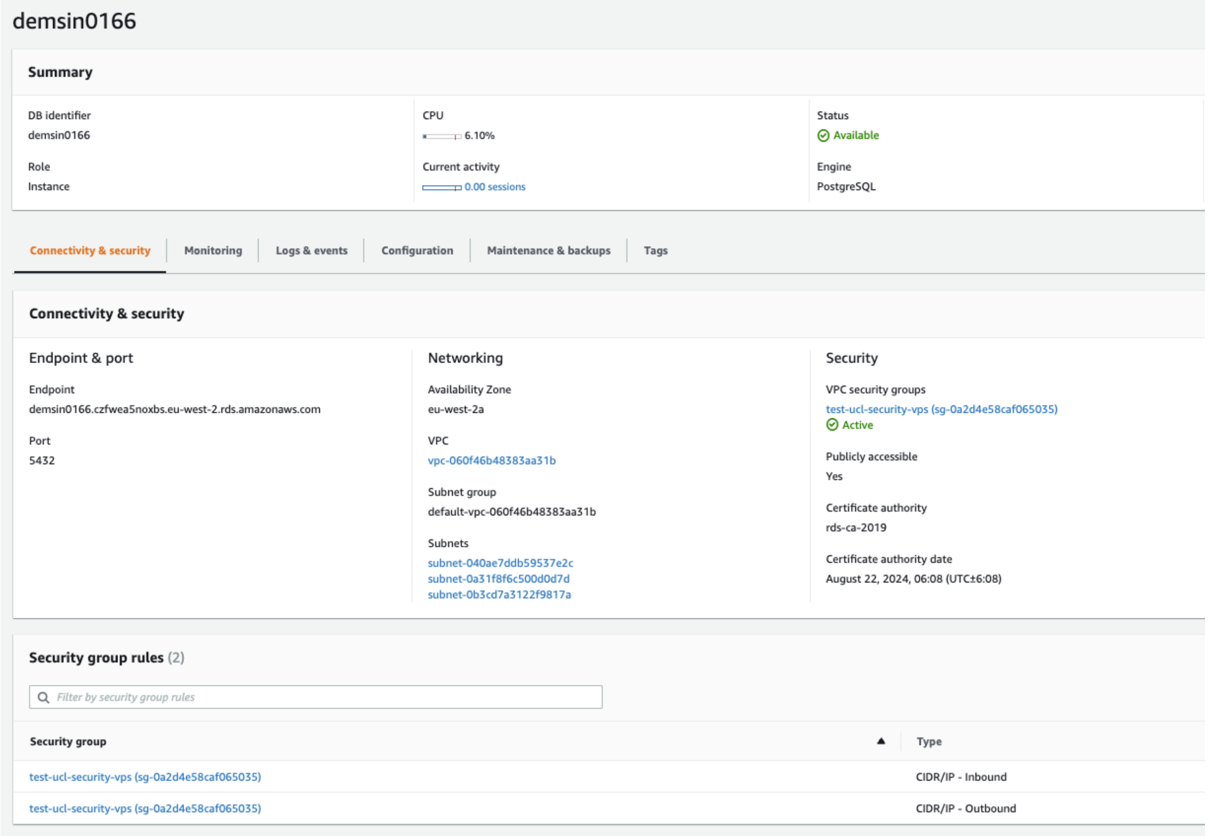
○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

Figure 2. Database Creation.

PostgreSQL version 12.7 was used for this project. The database was created using Amazon RDS leveraging their free tier allowance of 750 hours in a single instance and 20 GB of General-Purpose Storage (SSD) which was sufficient for this project, see figure 2.

After the database was initialized, user roles were created, and access was granted for each team member. Furthermore, as the DB instance is in a virtual private cloud, specific subnet and security groups had to be defined to ensure access for all users, see figure 3.



demsin0166

Summary

DB identifier	CPU	Status
demsin0166	6.10%	Available
Role	Current activity	Engine
Instance	0.00 sessions	PostgreSQL

Connectivity & security

Endpoint & port

Endpoint
demsin0166.czfwea5noxbs.eu-west-2.rds.amazonaws.com

Port
5432

Networking

Availability Zone
eu-west-2a

VPC
vpc-060f46b48383aa31b

Subnet group
default-vpc-060f46b48383aa31b

Subnets
subnet-040ae7ddb59537e2c
subnet-0a31f8f6c500d0d7d
subnet-0b3cd7a3122f9817a

Security

VPC security groups
test-ucl-security-vps (sg-0a2d4e58caf065035)
Active

Publicly accessible
Yes

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 06:08 (UTC±6:08)

Security group rules (2)

Security group	Type
test-ucl-security-vps (sg-0a2d4e58caf065035)	CIDR/IP - Inbound
test-ucl-security-vps (sg-0a2d4e58caf065035)	CIDR/IP - Outbound

Figure 3. Database Creation.

The main reason PostgreSQL, a relational database, was chosen over a NoSQL database such as Amazon's DynamoDB or MongoDB is because the goal of this project was to pull data from various sources and gather it in one place. Therefore, a relational database allows for creation of multiple tables that have relations and schemas which allows for more standardise, structured and organised data management. Whereas, a NoSQL database is schema-less, it has very few to no relations which is more suitable for storing unstructured data. The ER Diagram of the database can be seen in Figure 4.

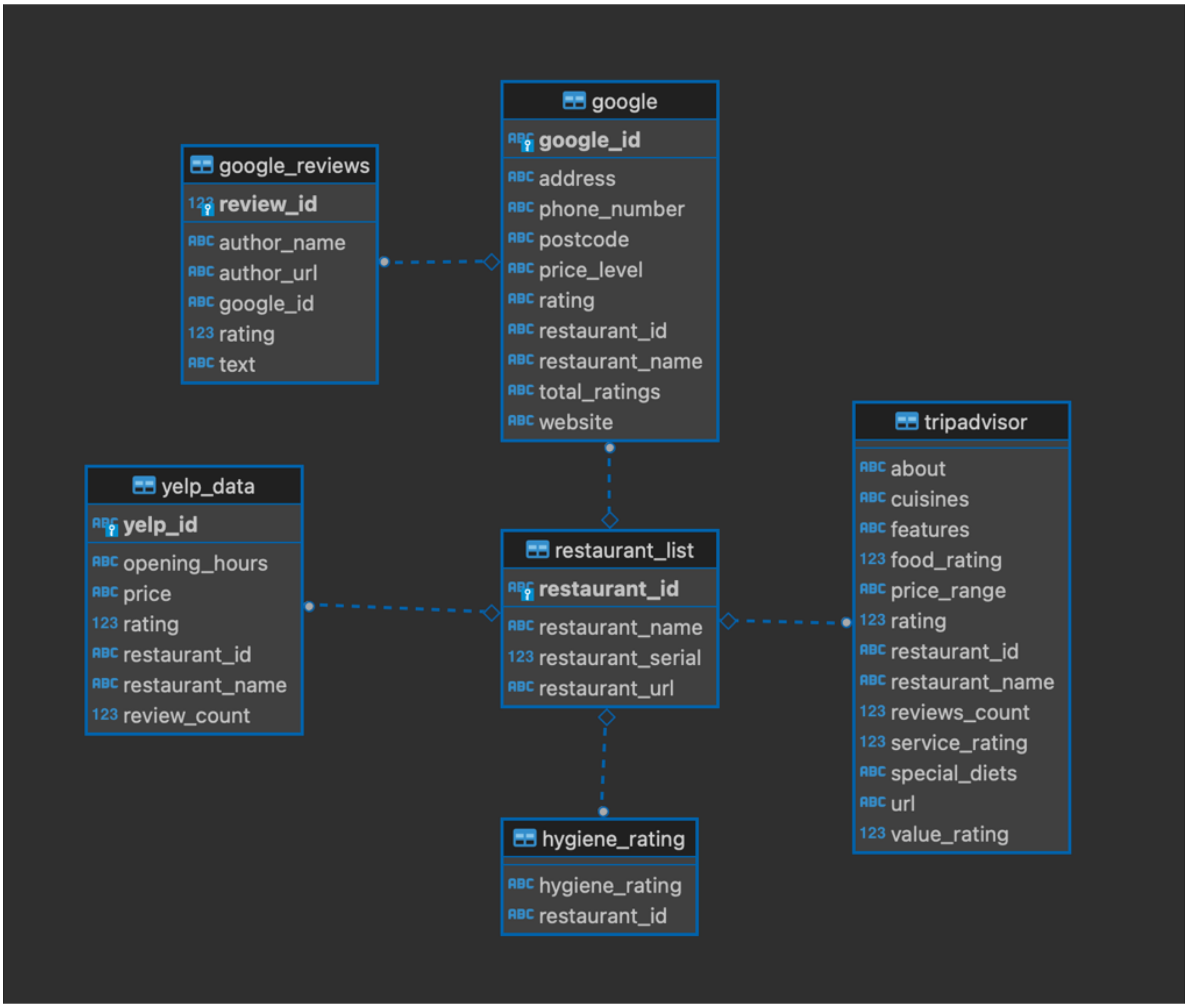


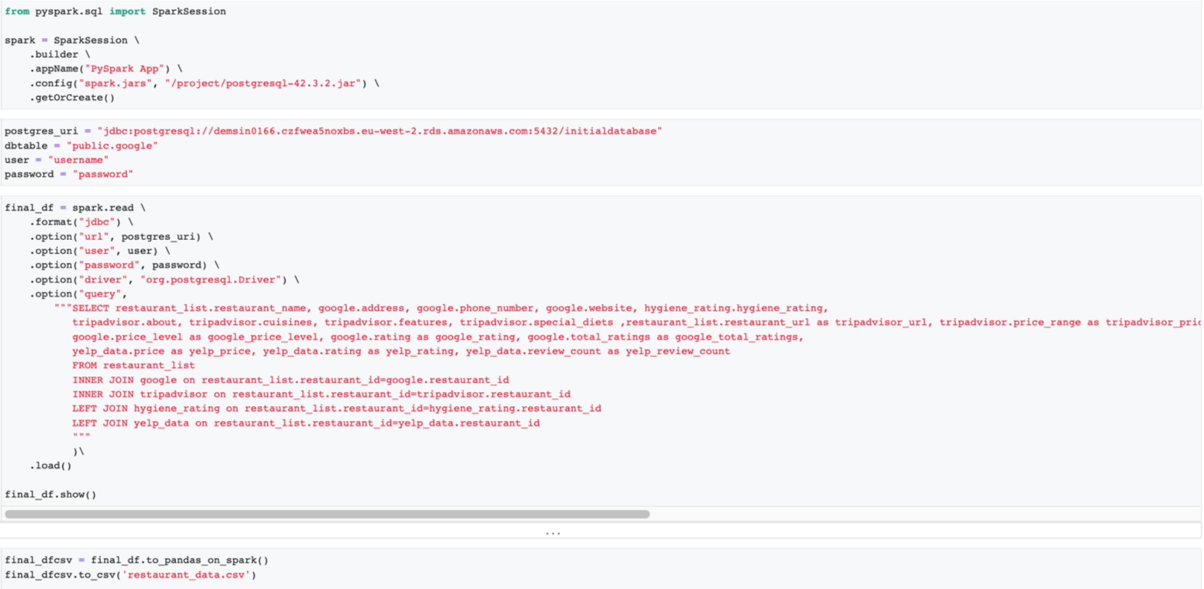
Figure 4. Database Schema.

2.3 Data Size and Complexity

This section will discuss the data size and complexity of the data selected. Apache Spark was used in this project to introduce a level of complexity and an element of adoption of an industry standard big data processing framework. Specifically, Pyspark was used, which is the Python API for Apache Spark that allows for large scale data processing. It has similar functionalities as Python's Pandas library, but it leverages Spark to allow for faster processing of large datasets through the distributed processing which also allows for better performance. Furthermore, Apache Spark itself is written in Scala and requires the user to be familiar with the Scala programming language, to use Spark. Therefore, Pyspark gives the ability to use Apache Spark with only the knowledge of Python.

Another large data processing framework is Apache Hadoop which uses the MapReduce algorithm to process data. The main difference between Apache Spark and Apache Hadoop is that Apache Hadoop reads and writes on the disk, whereas Apache Spark uses in-memory for storing data and operations which allows better performance and speed (Pointer, 2022). Moreover, Apache Spark includes additional components to the framework that are designated for use on large datasets on top of Spark Core, which is the foundation of the project. The supplementary components, include Spark SQL, Spark Streaming, MLib and GraphX.

Spark SQL is the module that allows the Spark and SQL integration, for structured data processing and querying data using SQL language. This was the main component used in this project, where Pyspark was used to convert files from Json to Parquet and then used to insert and query data from the Postgres database, see figure 5.



```
from pyspark.sql import SparkSession

spark = SparkSession \
    .builder \
    .appName("PySpark App") \
    .config("spark.jars", "/project/postgresql-42.3.2.jar") \
    .getOrCreate()
```

```
postgres_uri = "jdbc:postgresql://demsin0166.czfwea5noxbs.eu-west-2.rds.amazonaws.com:5432/initialdatabase"
dbtable = "public.google"
user = "username"
password = "password"
```

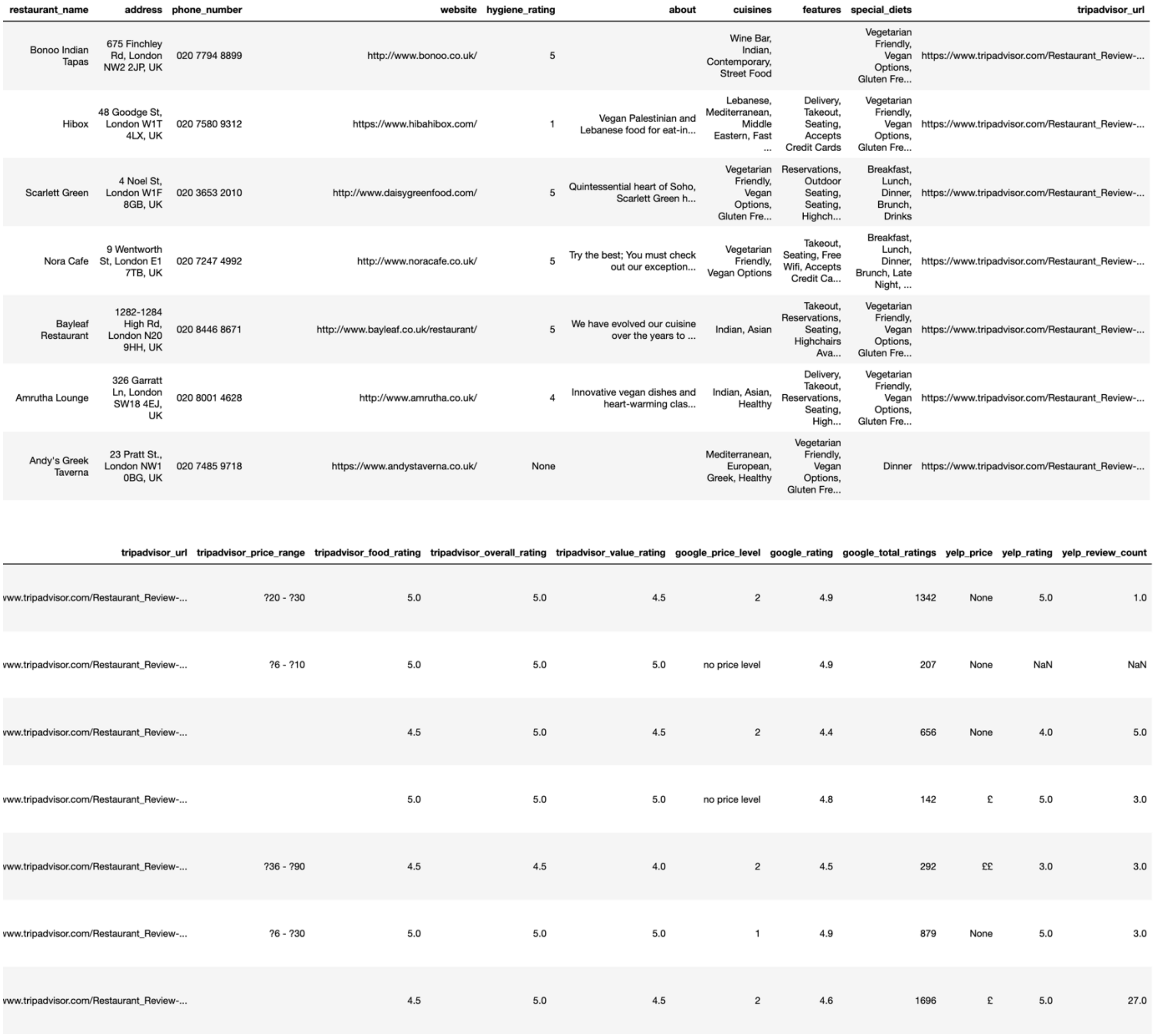
```
final_df = spark.read \
    .format("jdbc") \
    .option("url", postgres_uri) \
    .option("user", user) \
    .option("password", password) \
    .option("driver", "org.postgresql.Driver") \
    .option("query",
        """SELECT restaurant_list.restaurant_name, google.address, google.phone_number, google.website, hygiene_rating.hygiene_rating,
          tripadvisor.about, tripadvisor.cuisines, tripadvisor.features, tripadvisor.special_diets ,restaurant_list.restaurant_url as tripadvisor_url, tripadvisor.price_range as tripadvisor_pric
          google.price_level as google_price_level, google.rating as google_rating, google.total_ratings as google_total_ratings,
          yelp_data.price as yelp_price, yelp_data.rating as yelp_rating, yelp_data.review_count as yelp_review_count
          FROM restaurant_list
          INNER JOIN google on restaurant_list.restaurant_id=google.restaurant_id
          INNER JOIN tripadvisor on restaurant_list.restaurant_id=tripadvisor.restaurant_id
          LEFT JOIN hygiene_rating on restaurant_list.restaurant_id=hygiene_rating.restaurant_id
          LEFT JOIN yelp_data on restaurant_list.restaurant_id=yelp_data.restaurant_id
          """
          )\
    .load()

final_df.show()
```

...

```
final_dfcsv = final_df.to_pandas_on_spark()
final_dfcsv.to_csv('restaurant_data.csv')
```

Figure 5. Spark Session

After the database was completed and all the data was inserted, a final query was executed using Pyspark which merged all the tables to return a final DataFrame with the consolidated information. The Pyspark DataFrame was then converted to a Pandas DataFrame and saved as a CSV file for sharing and analytics, see figure 6 for a snapshot of the final output.

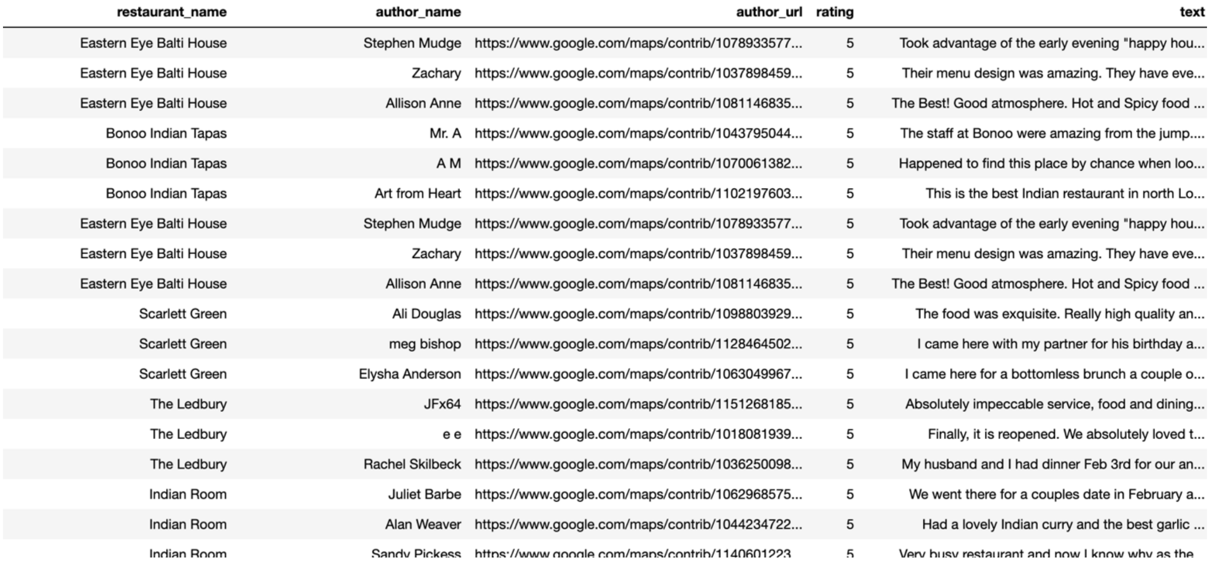


restaurant_name	address	phone_number	website	hygiene_rating	about	cuisines	features	special_diets	tripadvisor_url
Bonoo Indian Tapas	675 Finchley Rd, London NW2 2JP, UK	020 7794 8899	http://www.bonoo.co.uk/	5		Wine Bar, Indian, Contemporary, Street Food		Vegetarian Friendly, Vegan Options, Gluten Fre...	https://www.tripadvisor.com/Restaurant_Review-...
Hibox	48 Goodge St, London W1T 4LX, UK	020 7580 9312	https://www.hibahibox.com/	1	Vegan Palestinian and Lebanese food for eat-in...	Lebanese, Mediterranean, Middle Eastern, Fast ...	Delivery, Takeout, Seating, Accepts Credit Cards	Vegetarian Friendly, Vegan Options, Gluten Fre...	https://www.tripadvisor.com/Restaurant_Review-...
Scarlett Green	4 Noel St, London W1F 8GB, UK	020 3653 2010	http://www.daisygreenfood.com/	5	Quintessential heart of Soho, Scarlett Green h...	Vegetarian Friendly, Vegan Options, Gluten Fre...	Reservations, Outdoor Seating, Seating, Highch...	Breakfast, Lunch, Dinner, Brunch, Drinks	https://www.tripadvisor.com/Restaurant_Review-...
Nora Cafe	9 Wentworth St, London E1 7TB, UK	020 7247 4992	http://www.noracafe.co.uk/	5	Try the best; You must check out our exception...	Vegetarian Friendly, Vegan Options	Takeout, Seating, Free Wifi, Accepts Credit Ca...	Breakfast, Lunch, Dinner, Brunch, Late Night, ...	https://www.tripadvisor.com/Restaurant_Review-...
Bayleaf Restaurant	1282-1284 High Rd, London N20 9HH, UK	020 8446 8671	http://www.bayleaf.co.uk/restaurant/	5	We have evolved our cuisine over the years to ...	Indian, Asian	Takeout, Reservations, Seating, Highchairs Ava...	Vegetarian Friendly, Vegan Options, Gluten Fre...	https://www.tripadvisor.com/Restaurant_Review-...
Amrutha Lounge	326 Garratt Ln, London SW18 4EJ, UK	020 8001 4628	http://www.amrutha.co.uk/	4	Innovative vegan dishes and heart-warming clas...	Indian, Asian, Healthy	Delivery, Takeout, Reservations, Seating, High...	Vegetarian Friendly, Vegan Options, Gluten Fre...	https://www.tripadvisor.com/Restaurant_Review-...
Andy's Greek Taverna	23 Pratt St., London NW1 0BG, UK	020 7485 9718	https://www.andystaverna.co.uk/	None		Mediterranean, European, Greek, Healthy	Vegetarian Friendly, Vegan Options, Gluten Fre...	Dinner	https://www.tripadvisor.com/Restaurant_Review-...

tripadvisor_url	tripadvisor_price_range	tripadvisor_food_rating	tripadvisor_overall_rating	tripadvisor_value_rating	google_price_level	google_rating	google_total_ratings	yelp_price	yelp_rating	yelp_review_count
www.tripadvisor.com/Restaurant_Review-...	?20 - ?30	5.0	5.0	4.5	2	4.9	1342	None	5.0	1.0
www.tripadvisor.com/Restaurant_Review-...	?6 - ?10	5.0	5.0	5.0	no price level	4.9	207	None	NaN	NaN
www.tripadvisor.com/Restaurant_Review-...		4.5	5.0	4.5	2	4.4	656	None	4.0	5.0
www.tripadvisor.com/Restaurant_Review-...		5.0	5.0	5.0	no price level	4.8	142	£	5.0	3.0
www.tripadvisor.com/Restaurant_Review-...	?36 - ?90	4.5	4.5	4.0	2	4.5	292	££	3.0	3.0
www.tripadvisor.com/Restaurant_Review-...	?6 - ?30	5.0	5.0	5.0	1	4.9	879	None	5.0	3.0
www.tripadvisor.com/Restaurant_Review-...		4.5	5.0	4.5	2	4.6	1696	£	5.0	27.0

Figure 6. Final Output

Additionally, a separate table with just Google reviews was queried and saved to a CSV file, to avoid duplicate rows in the main file, see figure 6. The reviews file can then be used to merge to the main file as needed, to allow for supplementary analytics.



restaurant_name	author_name	author_url	rating	text
Eastern Eye Balti House	Stephen Mudge	https://www.google.com/maps/contrib/1078933577...	5	Took advantage of the early evening "happy hou...
Eastern Eye Balti House	Zachary	https://www.google.com/maps/contrib/1037898459...	5	Their menu design was amazing. They have eve...
Eastern Eye Balti House	Allison Anne	https://www.google.com/maps/contrib/1081146835...	5	The Best! Good atmosphere. Hot and Spicy food ...
Bonoo Indian Tapas	Mr. A	https://www.google.com/maps/contrib/1043795044...	5	The staff at Bonoo were amazing from the jump....
Bonoo Indian Tapas	A M	https://www.google.com/maps/contrib/1070061382...	5	Happened to find this place by chance when loo...
Bonoo Indian Tapas	Art from Heart	https://www.google.com/maps/contrib/1102197603...	5	This is the best Indian restaurant in north Lo...
Eastern Eye Balti House	Stephen Mudge	https://www.google.com/maps/contrib/1078933577...	5	Took advantage of the early evening "happy hou...
Eastern Eye Balti House	Zachary	https://www.google.com/maps/contrib/1037898459...	5	Their menu design was amazing. They have eve...
Eastern Eye Balti House	Allison Anne	https://www.google.com/maps/contrib/1081146835...	5	The Best! Good atmosphere. Hot and Spicy food ...
Scarlett Green	Ali Douglas	https://www.google.com/maps/contrib/1098803929...	5	The food was exquisite. Really high quality an...
Scarlett Green	meg bishop	https://www.google.com/maps/contrib/1128464502...	5	I came here with my partner for his birthday a...
Scarlett Green	Elysha Anderson	https://www.google.com/maps/contrib/1063049967...	5	I came here for a bottomless brunch a couple o...
The Ledbury	JFx64	https://www.google.com/maps/contrib/1151268185...	5	Absolutely impeccable service, food and dining...
The Ledbury	e e	https://www.google.com/maps/contrib/1018081939...	5	Finally, it is reopened. We absolutely loved t...
The Ledbury	Rachel Skilbeck	https://www.google.com/maps/contrib/1036250098...	5	My husband and I had dinner Feb 3rd for our an...
Indian Room	Juliet Barbe	https://www.google.com/maps/contrib/1062968575...	5	We went there for a couples date in February a...
Indian Room	Alan Weaver	https://www.google.com/maps/contrib/1044234722...	5	Had a lovely Indian curry and the best garlic ...
Indian Room	Sandy Pickess	https://www.google.com/maps/contrib/1140601223	5	Very busy restaurant and now I know why as the

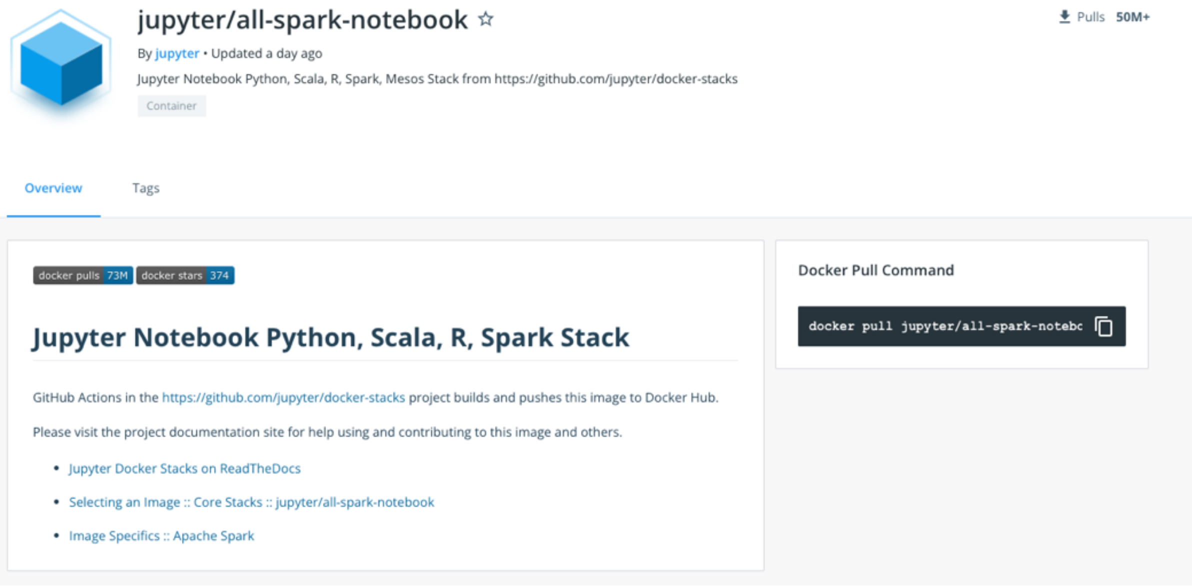
Figure 7. Google Reviews

3.0 Reproducibility and Version Control

3.1 Docker

Docker was used to ensure that Spark could be run by each user. It provides the ability to containerise all the dependencies and requirements needed to run an application and share between the team, so anyone could run the application without the need to manually install different pre-requisites. In this case, the aim was to run a Jupyter Notebook with Spark. Without Docker, multiple steps are required to achieve the same objective on macOS and Linux, such as installing homebrew for package management, installing Xcode which is required for installing other dependencies, then installing Java, Scala, and Spark individually. Finally, the installation must be verified by running Spark to ensure that it has been successfully installed.

As can be seen, many steps are required to run Spark which also means that each step can be prone to errors due to versioning or other issues. This results in difficulties and high time consumption, especially for inexperienced users. Therefore, Docker simplifies this process, where the user only needs to install Docker Desktop app and execute a Dockerfile script which automatically builds an image and then allows the user to run a container. For this project, only Spark was needed. Hence, the only steps required were to pull a pre-build image 'all-spark-notebook' created by the Jupyter Development Team from the dockerhub and run a simple container with specified local host to successfully run a Jupyter Notebook with Pyspark. To further simplify this process and make it shareable with others, a docker-compose.yml file was created, so the only step required is to run a docker-compose up command in the terminal to download the specific image and initialise the container in one procedure. The file was than shared in the GitHub repository.



Jupyter Notebook Python, Scala, R, Spark, Mesos Stack from https://github.com/jupyter/docker-stacks

Jupyter Notebook Python, Scala, R, Spark Stack

GitHub Actions in the https://github.com/jupyter/docker-stacks project builds and pushes this image to Docker Hub.

Please visit the project documentation site for help using and contributing to this image and others.

- Jupyter Docker Stacks on ReadTheDocs
- Selecting an Image :: Core Stacks :: jupyter/all-spark-notebook
- Image Specifics :: Apache Spark

```
docker pull jupyter/all-spark-notebc
```

An alternative to Docker is to use virtual machines, which require an entire OS to be loaded, to ensure that installing different dependencies will not affect the base system of the user. Hence, it is time-consuming and costly as it requires more memory and resources which often affects the performance compared to Docker. External services such as AWS or even Faculty can be used to run VM in cloud but that would incur extra expenses. Furthermore, Docker is gaining traction with millions of users and many large and small companies are using it.

3.2 GitHub

GitHub was chosen for our version control and file share tools for three reasons. First, it is widely used in both industry and academic, and there are many guides and resources on the internet. Second, GitHub Desktop makes it simple and easy to compare codes between different versions. Third, GitHub provides unlimited public repository so that it is easy to share and showcase this project. During this project, file sharing and code comparing are the top two frequently used functions. Moreover, local backups are used with GitHub to ensure all versions of files are kept.

The file is structured as datafile, script, and report. The script can be further divided by names of different tools. The link is below: https://github.com/uceis42/Data-Engineering-2022-Groupwork

4.0 Project Management

In terms of managing the project, weekly meetings were held on Fridays to discuss each member's progress. Each meeting lasted for an hour and thirty minutes. The project was divided based on websites (for example, one group member was responsible for extracting the data from Google) for the data collection phase. Project Management software such as Notion was also used to keep track of both individual and groups deadlines. Microsoft Teams was used to hold meetings and update the shared Word document for the written report.

Each meeting focused on a different target—the first meeting involved coming up with feasible project ideas and a proposed structure. The next meeting involved sharing each restaurant-based website to a team member with each team member extracting the required information from the respective websites. The third meeting involved discussing and creating the appropriate database schemas and relationships between each member’s table. The final meeting focused on writing up and editing the final report.

5.0 Real world application with the dataset

Our dataset can be used in many real-world applications by different industries or in studies and research. Firstly, combined with an in-house user information dataset, this dataset can be adopted in a recommendation system for local lifestyle apps. The matching between the user’s profile and interest and the restaurant’s cuisine, location, and price range will indicate a reasonable recommendation.

Secondly, this dataset can be also used in marketing research on restaurants. For example, it can be used to perform a location analysis for a new local restaurant. With the help of map information, our dataset can indicate the concentration level of certain cuisines in specific areas as well as the price range level. The incoming local restaurant can make a better decision based on these results.

Thirdly, a food culture study on cuisine over areas can be carried out by our dataset. Since this dataset contains both google review data for sentiment analysis and cuisine over location data, a food culture study can be developed. Additionally, academic studies on media influence can be performed based on our dataset. Most restaurants have different ratings and reviews on different platforms. Since we gather information from four different data sources, and there are many overlapping restaurants, the comparison across platforms will show how media influence people's choices and opinions.

Lastly, our dataset can help in the public health area. This dataset has a food hygiene rating with all other information like location, price, and special diets. All these data can be integrated into an application that shows the hygiene level of a specific restaurant or the entire area, which can improve the public awareness of food safety and the hygiene level of the local areas.

6.0 Summary and Limitations

In Summary, there are mainly three parts in this project: 1. gathering data by APIs and web scraping. 2. Transforming data to the parquet format, then import to our AWS database. 3. Combining and joining all data from the database as one final dataset for future use. During these processes, GitHub is utilized for version control, and Docker is used for easy reproducibility. Finally, the dataset is generated by joining cleaned data from different sources.

The limitation for this project is that we did not include time in our dataset, so that there is no comparison between before and after some time period. Similarly, due to the limitation of time and resources, the data size is limited to a level of 200-300 rows.

Further improvement for this project can be mainly focus on enlarge the size of dataset, as well as adding time as a variable. The overall structure for the entire project can remain unchanged.

7.0 References

1. Apache Spark – Wikipedia. Apache Spark - Wikipedia (2022). Available at: https://en.wikipedia.org/wiki/Apache_Spark (Accessed: 1 April 2022).

2. GraphQL [online] available from https://www.howtographql.com/basics/1-graphql-is-the-better-rest/ [17th March 2022]

3. IBM Education. (2021) What is Docker?, Ibm.com. Available at: https://www.ibm.com/cloud/learn/docker (Accessed: 1 April 2022).

4. Ivanov, T. and Pergolesi, M., 2020. The impact of columnar file formats on SQL-on-hadoop engine performance: A study on ORC and Parquet. Concurrency and Computation: Practice and Experience, 32(5), p.e5523.

5. Levy, E. (2022) 'What is the Parquet File Format and Why You Should Use It.' Available at: https://www.upsolver.com/blog/apache-parquet-why-use
6. Lomborg, S. and Bechmann, A., 2014. Using APIs for data collection on social media. The Information Society, 30(4), pp.256-265. Available from https://www.tandfonline.com/doi/pdf/10.1080/01972243.2014.915276?casa_token=36upPlRaucoAAAAA:itISyy0cLDP151T2UGSsByCKdhCWXVRdgmitqR3mxkkqeXhxwLk01Yk8ibbyqHv63FUvPmigMec [17th March 2022]

7. Pointer, I. (2022) What is Apache Spark? The big data platform that crushed Hadoop, InfoWorld. Available at: https://www.infoworld.com/article/3236869/what-is-

apache-spark-the-big-data-platform-that-crushed-hadoop.html (Accessed: 1 April 2022).

8. SQL vs NoSQL comparison: MySQL, PostgreSQL, MongoDB & Cassandra (2021). Available at: https://devathon.com/blog/sql-vs-nosql-mysql-vs-postgresql-vs-mongodb-vs-cassandra/#:~:text=PostgreSQL%20stores%20structured%20data.,it%20can%20store%20unstructured%20data. (Accessed: 31 March 2022).

9. What is Spark SQL? - Databricks (2022). Available at: https://databricks.com/glossary/what-is-spark-sql#:~:text=Spark%20SQL%20is%20a%20Spark,on%20existing%20deployments%20and%20data. (Accessed: 1 April 2022).