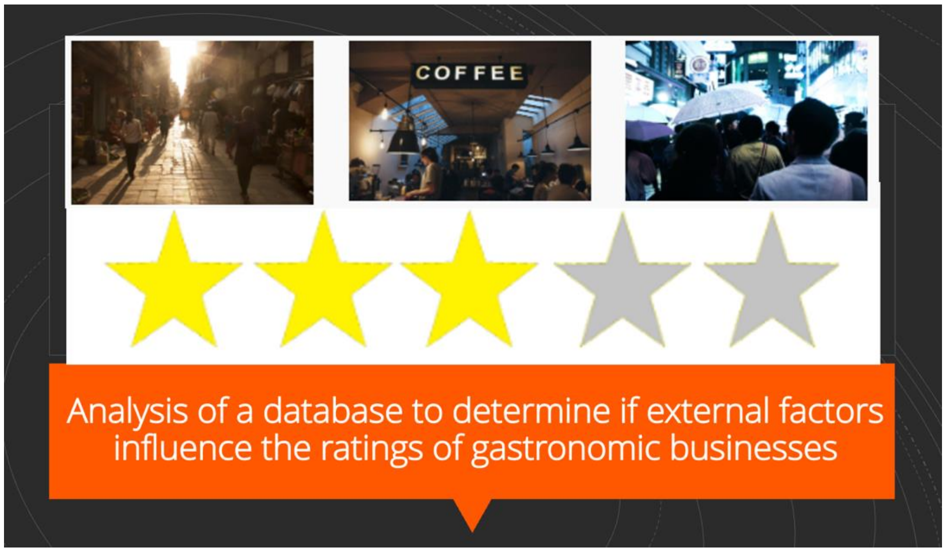
HSLU

Database Management for Data Scientists
DBM02.H2101 Module – Semester Project: Technical Report


COFFEE
Analysis of a database to determine if external factors influence the ratings of gastronomic businesses

Group: The Team

Name	E-Mail
Alina Eckstein	Alina.eckstein@stud.hslu.ch
Hendrik Croce	Hendrik.croce@stud.hslu.ch
Hossam Fahmy	Hossameldin.fahmy@stud.hslu.ch
Nathan Rohrbach	Nathan.rohrbach@stud.hslu.ch

1 Context and Motivation

1.1 Introduction

The gastronomy industry is in a dynamic environment. Due to the ever-increasing use of smartphones and apps, gastronomy customers have changed their behavior. Nowadays often, people read and inform themselves about the reviews of gastronomy businesses before visiting or booking. Therefore, it is crucial for business owners and investors to have good ratings and, accordingly, to know the reasons behind negative reviews to make the necessary improvements. However, not all factors which could have a negative impact on the customer experience can be changed, but only anticipated and, in the best case, taken into account.

The hypothesis for this project is based on the argument that customer experience, and thus gastronomy ratings, might also be influenced by external factors such as weather, type of neighborhood or accessibility by public transport, for example. Therefore, we want to examine the relationship between gastronomy business ratings and weather conditions. For this, we will merge two large, real-world datasets to draw conclusions about how weather affects Yelp reviews.

Yelp Inc is a US publicly listed company and operates one of the biggest recommendation portals. Yelps business model focuses on user-generated reviews and recommendations for service providers such as restaurants and bars.

1.2 Usability

With the SQL database and the resulting visualizations that we developed, we want to provide a tool to advise restaurateurs and investors.

It enables not only to demonstrate whether climate and weather affect their business success rate but can also be used as a general source to gain information about existing businesses, their rating and location, to carry out market and/or competitor analyses or to investigate weather conditions and find attractive locations for the establishment of new gastronomic businesses.

1.3 Limitations

As mentioned in our hypothesis, there can be many different external factors that potentially influence customer experience and ratings. In this project we focus specifically on one external factor, namely the weather. Furthermore, the analysis will be performed based on a time period of 10 years (2008-2017) and five cities in the United States (Cleveland, Charlotte, Las Vegas, Phoenix and Pittsburgh).

1.4 Additional business requirements

Ownership of data

The Team consisting of four members will had read and write access to the data.

Who will have access

Future customers, will have read only access with no permission to alter the data.

Scalability

Once we create the first database, we expect to add 20% more of data per year over the next 5 years. We envision to include further data source such as crime rates and other external factors.

2 Use Case and Data Selection

2.1 Use Case Definition

Online reviews sites like Yelp and Google are vital for many gastronomy businesses. Through a few good ratings a hype can arise. But the same thing can happen in the other direction with a few bad ones.

A study in the USA with 32 restaurants showed that three factors lead to bad customer reviews: bad service, bad food and bad weather. Restaurant managers can influence the first two arguments. But the latter, namely the weather, they cannot.[1]

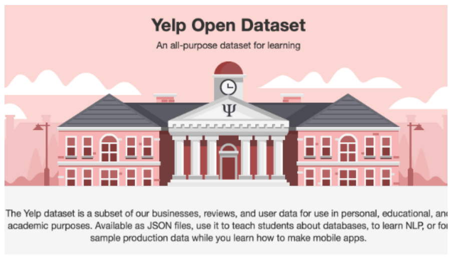
Our database management project aims to analyse how the ratings of gastronomic business in the USA are influenced by the external factor of weather.

2.2 Data Source Selection

To be able to perform the analysis we carefully selected two data sources:

Data source 1: Yelp

The first dataset is an open dataset from Yelp that provides us with data around the businesses, reviews, user data. The dataset contains in total more than 1'600'000 businesses and many more reviews. We will take a subset of this dataset with specific information about the restaurants in the location of our interest.

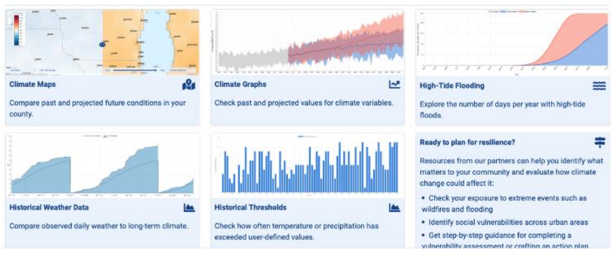


The Yelp dataset is a subset of our businesses, reviews, and user data for use in personal, educational, and academic purposes. Available as JSON files, use it to teach students about databases, to learn NLP, or for sample production data while you learn how to make mobile apps.

Figure 1 Data source 1

Data source 2: Climate

The second dataset contains climate data from the National Environmental Modeling and Analysis Center (NEMAC). From this data source we will retrieve the corresponding temperature and precipitation data in the locations where our analyzed businesses are situated.



Climate Maps

Compare past and projected future conditions in your county.

Climate Graphs

Check past and projected values for climate variables.

High-Tide Flooding

Explore the number of days per year with high-tide floods.

Historical Weather Data

Compare observed daily weather to long-term climate.

Historical Thresholds

Check how often temperature or precipitation has exceeded user-defined values.

Ready to plan for resilience?

Resources from our partners can help you identify what matters to your community and evaluate how climate change could affect it:

- Check your exposure to extreme events such as wildfires and flooding
- Identify social vulnerabilities across urban areas
- Get step-by-step guidance for completing a vulnerability assessment or crafting an action plan

Figure 2 Data source 2

[1] https://www.sciencedaily.com/releases/2019/04/190422082243.htm

2.3 Data Source Selection

In our project, we went through the following process. First, we obtained the two data sources with the different data files. Then we prepared and merged the datasets and created one MySQL database. In the end we fed python and tableau with our MySQL database and consumed the information by creating python analysis and tableau visualizations.

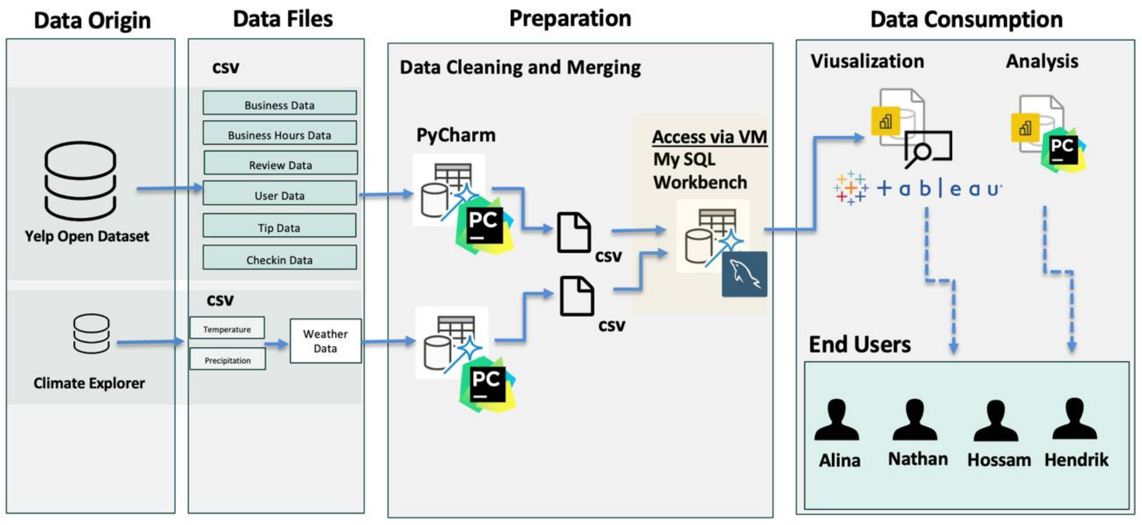


Figure 3 Project Process

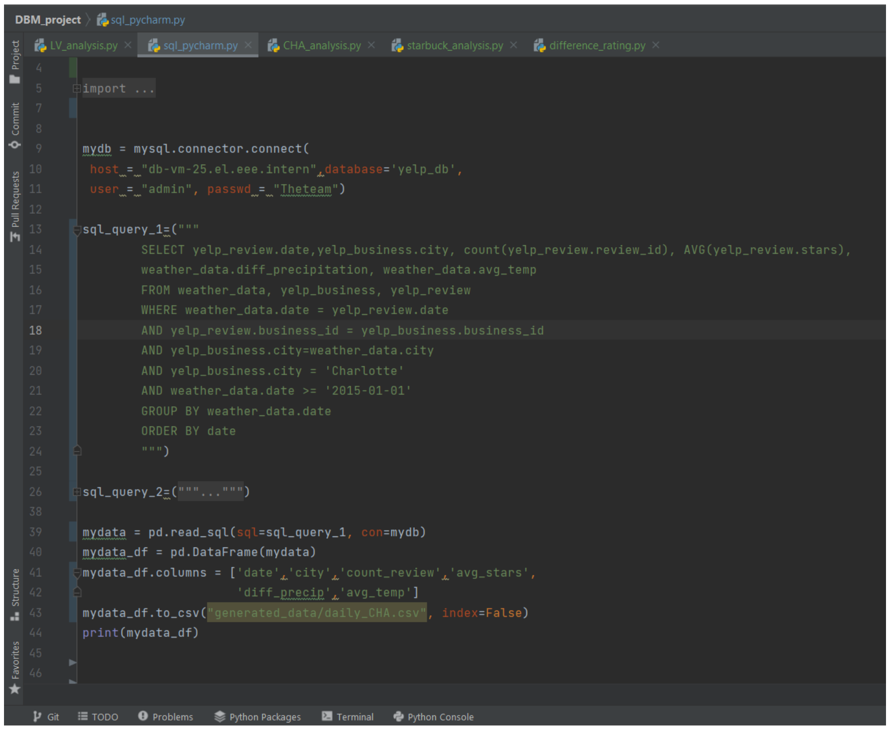
3 Database Implementation

In order for everyone in the team to be able to run a database locally, we have each installed the MySQL Workbench and the MySQL Community Server from Oracle. Thus, all members of the team were able to work independently and locally. Subsequently, a SQL script was written for creating the database based on the entity relation model.

3.1 Choice of Database Technology

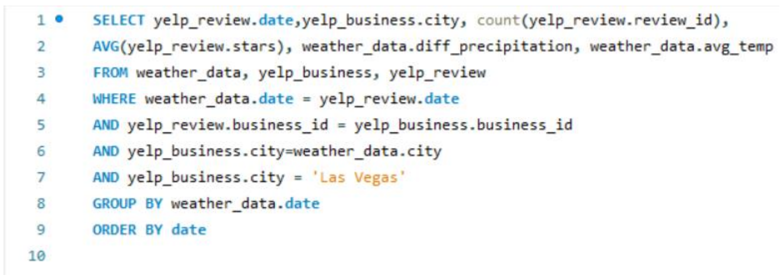
As database language we choose an SQL over a NoSQL approach. One of the main reasons for that choice is the incorporation of an Entity-Relationship-Model. The database we have chosen fits perfectly for that approach since the relations between the variables play an important role for our query. In addition, it is stable and allows straight forward query processing. Eventually SQL-knowledge is a demanded skill as a data scientist which was an incentive as well. As database system we have used MySQL.

For a meaningful analysis and result, we used 'mysql.connector' to connect SQL Server to PyCharm and create a dataset from SQL queries (see section 5).



```python
import ...


mydb = mysql.connector.connect(
 host = "db-vm-25.el.eee.intern",database='yelp_db',
 user = "admin", passwd = "Theteam")

sql_query_1=("""
        SELECT yelp_review.date,yelp_business.city, count(yelp_review.review_id), AVG(yelp_review.stars),
        weather_data.diff_precipitation, weather_data.avg_temp
        FROM weather_data, yelp_business, yelp_review
        WHERE weather_data.date = yelp_review.date
        AND yelp_review.business_id = yelp_business.business_id
        AND yelp_business.city=weather_data.city
        AND yelp_business.city = 'Charlotte'
        AND weather_data.date >= '2015-01-01'
        GROUP BY weather_data.date
        ORDER BY date
        """)

sql_query_2=("""...""")

mydata = pd.read_sql(sql=sql_query_1, con=mydb)
mydata_df = pd.DataFrame(mydata)
mydata_df.columns = ['date','city','count_review','avg_stars',
                     'diff_precip','avg_temp']
mydata_df.to_csv("generated_data/daily_CHA.csv", index=False)
print(mydata_df)
```

Figure 4 SQL Pycharm.py



```sql
SELECT yelp_review.date,yelp_business.city, count(yelp_review.review_id),
AVG(yelp_review.stars), weather_data.diff_precipitation, weather_data.avg_temp
FROM weather_data, yelp_business, yelp_review
WHERE weather_data.date = yelp_review.date
AND yelp_review.business_id = yelp_business.business_id
AND yelp_business.city=weather_data.city
AND yelp_business.city = 'Las Vegas'
GROUP BY weather_data.date
ORDER BY date
```

Example of sql query used in pycharm to get the desired dataset from the database.

(See section 5. Analysis to see the processed information from this table)

Figure 5 exe SQL 1

3.2 System Architecture

Although we were able to use a Virtual Machine to run all processes and store the data, we used rather an alternative approach which is GitHub[2] for working together. On the file hosting service, we saved the necessary scripts as well as the project description and a constantly updated the technical report. The dataset itself was stored on each of the team members local drive, in order to maximize security. Another point we took into consideration is the simple fact that the admin would have had to restart the Virtual Machine every day just needed to much effort and would not be ecologically sustainable.

[2] NatchosR/Restaurants-Database-in-the-US (github.com)

3.3 Data

From yelp we have 6 files and from Climate Explorer 2 files. We merged the Climate data into one Weather data file as seen in Figure 6 Sources.

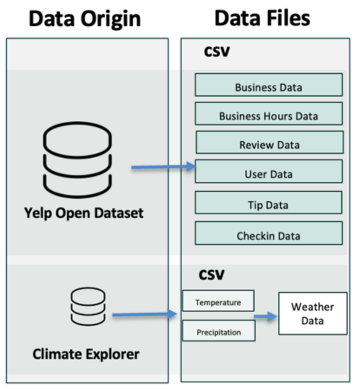


Figure 6 Sources

The data was in a csv format and wasn't normalized. The data sets have information about businesses and their respective details (opening hours) as well as information on users.

Sensitivity of data

All the data we had is public information and doesn't have any sensitive information such as personal and family name of the users but instead username is used.

As the file size in the raw format was very large, we had to do some cleaning steps in order to use our data properly. We had data going back to early 2000's which was too much for us to analyse and make conclusions. Therefore, we decided to only use data from 2008 onwards.

Further we took the average daily temperature as we were given maximum and minimum temperatures, and we used the differential daily precipitation as we were given the accumulated precipitation values on each day. This way we have a clearer values and information on weather data.

File	file type	Raw Size [MB]	Tables	Row; Column	Cleaned Size [MB]
yelp_tip.csv	csv	141	yelp_tip*	1112819; 5	145.7
yelp_user.csv	csv	1260	yelp_user	1268697; 5	123.8
yelp_review.csv	csv	3530	yelp_review	1737342; 8	616
yelp_business.csv	csv	290	yelp_business	19376; 12	47.2
yelp_business_hours.csv	csv	13.2	yelp_business_hours*	173408; 8	19.5
yelp_checkin.csv	csv	129	yelp_checkin*	3896402; 4	233.8
weather_data.csv	csv	1.39	weather_data	18360; 13	2.5

* Table not processed in the database cleaning. Efficiency was not necessary on those table as they were not central table.

3.4 Normalization

To create our data model we started with the most general level of data modeling. At the conceptual level, we thought of creating entities that represent business objects for the database. Here we thought broadly. The next refinement to the conceptual ERD is when we added attributed in the ERD with human-friendly entity and attribute names. In addition, the created relationship lines as required. In the last step we created the physical data Figure 8 Final ERD which was built in the database. Each entity should represent a database table, complete with column names and data types. In addition, Primary keys and Foreign keys have been assigned. With the final step we developed a relational database set to the 3NF.

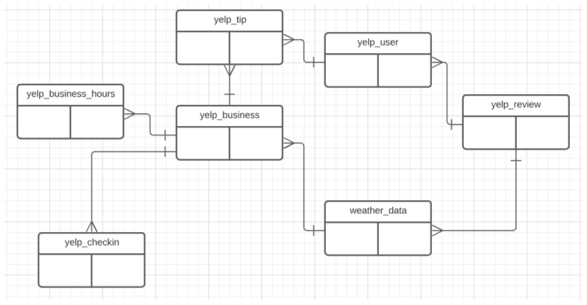


Figure 7 Conceptual ERD

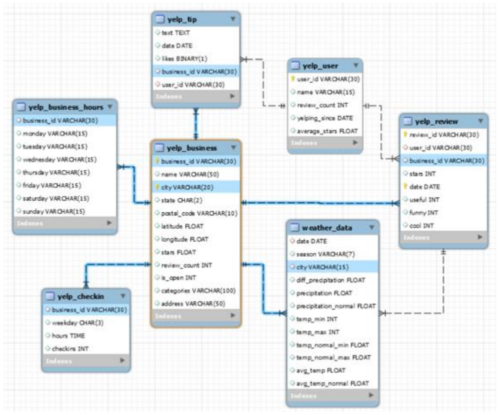


Figure 8 Final ERD

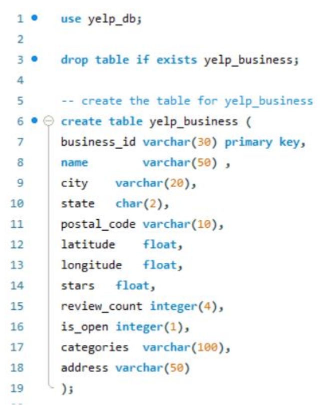
3.5 Data Import

First, we created the database, yelp_db, with the reverse engineer features. We made a file with all tables and their variables with the data type. Thanks to this method, we were able to develop the database in MySQL Workbench (schema).

Unfortunately, due to the size of each csv-file, the loading process failed. Therefore, we had to change our strategy to load the data into the database. We created individual SQL files, where we generated the table in the database yelp_db and then loaded the data from the csv-file in the corresponding column. This technique allowed us to have a greater control on the column variable and thus, successfully load large csv files (~3.5GB) into the database on MySQL workbench.

<table>
<tr>
<td>

1)



```
use yelp_db;

drop table if exists yelp_business;

-- create the table for yelp_business
create table yelp_business (
business_id varchar(30) primary key,
name        varchar(50) ,
city    varchar(20),
state   char(2),
postal_code varchar(10),
latitude    float,
longitude   float,
stars   float,
review_count integer(4),
is_open integer(1),
categories  varchar(100),
address varchar(50)
);
```

Figure 9 exe SQL 2

</td>
<td>

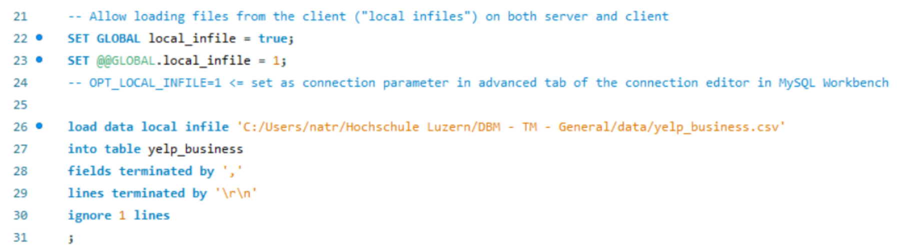
1) Implement the table in the database yelp_db. Then create the table with the variable and the corresponding data type.

2) Then comes the loading phase, with the first commands that allow the client to load files into the database. (Then we disabled this option for security reason).

Note:
The data type is extremely important when loading large files. A small error, like not enough digits or characters can lead to failure that will cost precious memory space in the Virtual Machine and is extremely time consuming (loading a file of 3.5GB takes around 10min).

</td>
</tr>
</table>

2)



```
-- Allow loading files from the client ("local infiles") on both server and client
SET GLOBAL local_infile = true;
SET @@GLOBAL.local_infile = 1;
-- OPT_LOCAL_INFILE=1 <= set as connection parameter in advanced tab of the connection editor in MySQL Workbench

load data local infile 'C:/Users/natr/Hochschule Luzern/DBM - TM - General/data/yelp_business.csv'
into table yelp_business
fields terminated by ','
lines terminated by '\r\n'
ignore 1 lines
;
```

Figure 10 exe SQL 3

There are seven tables in the database, so we executed this process seven times. Some tables had a very complex structure and required cleaning operations before loading them into the database. For instance, yelp_review.csv has a column of large text (customer reviews). Due to the size of csv-file, we could not process it in Excel or PyCharm (too long processing time). Therefore, we used the Linux Terminal to process those cleaning actions. For other tables, PyCharm worked fine.

Once all the data was loaded in the database, we linked the tables with primary and foreign key in MySQL Workbench.

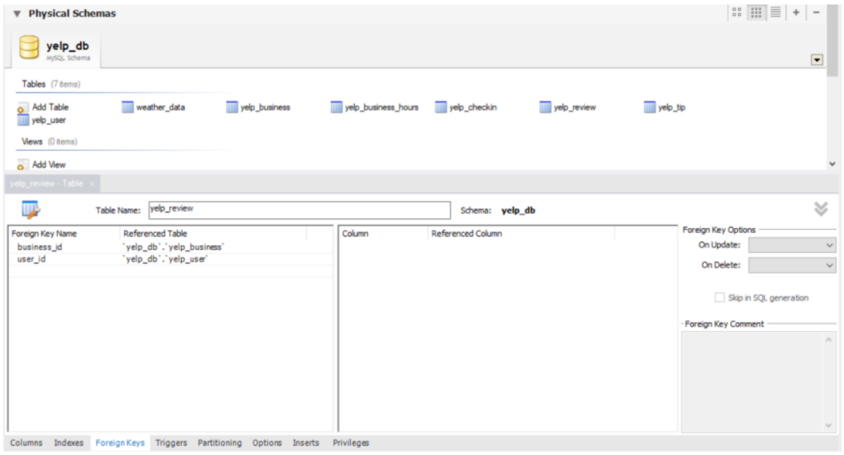


Table Name: yelp_review Schema: yelp_db

Foreign Key Name	Referenced Table
business_id	`yelp_db`.`yelp_business`
user_id	`yelp_db`.`yelp_user`

Column	Referenced Column

Foreign Key Options
On Update:
On Delete:
Skip in SQL generation
Foreign Key Comment

Figure 11 exe SQL 4

3.6 Data Schema Model

Below, the database, yelp_db, with all connected tables is shown. This schema has two main tables, yelp_business and yelp_review. This structure allows the user to access all the data in less than three table connections, which keeps the queries simple and effective. Most of the connections go through business_id, date and city, to access weather, reviews and business data. We strategically chose those keys so we can add other tables in the database for further analysis. For instance, if we want to analyse the effect of crime rates in a city on the business reviews, we can easily download the data set and connect it to the key date and city. A more complex feature would be to include transportation data. To do so we need the address of each business so we can link a bus stop or a train station. Fortunately, the address of each business is included in the business table. As one can see, the database has multiple improvement possibilities.

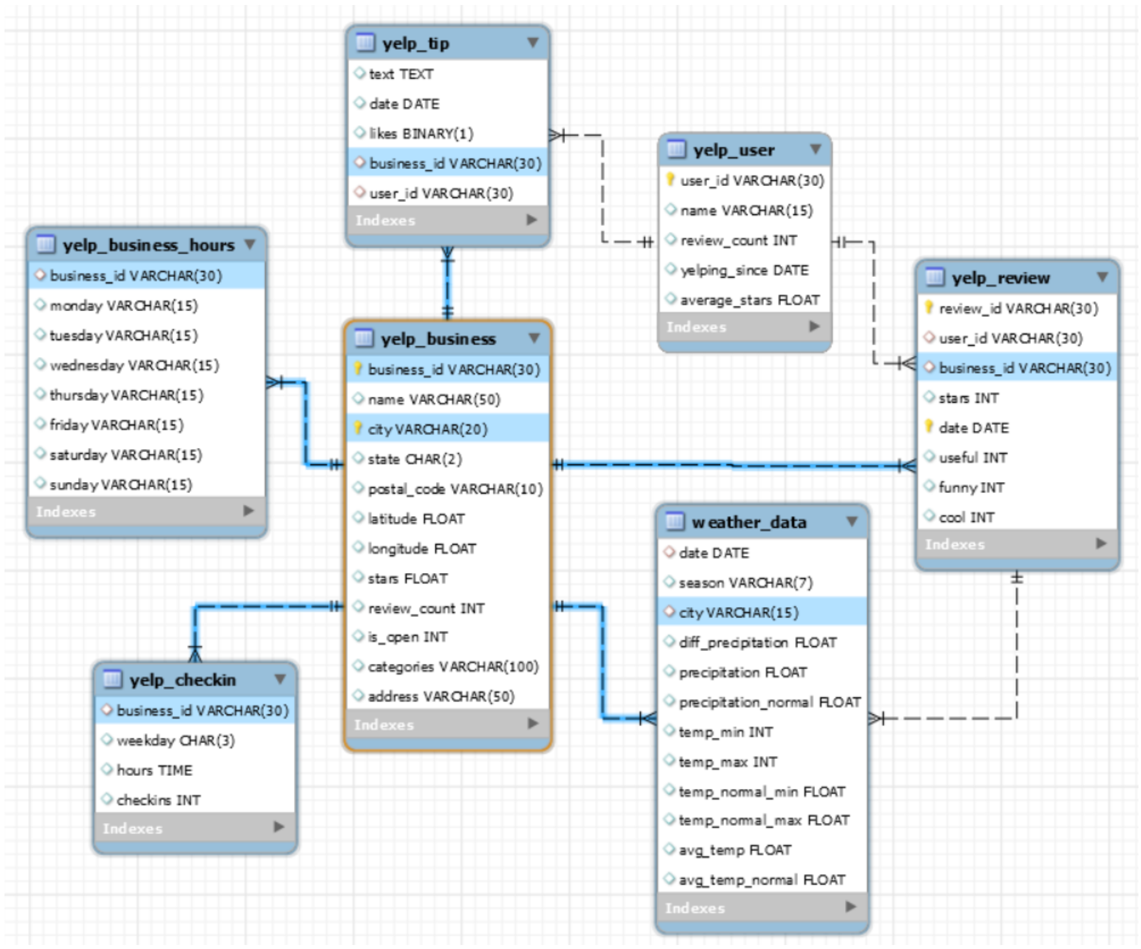


Figure 12 ERD

3.7 Efficiency Optimization

The database is 11GB large, and the main table 'yelp_reviews' contains 3.5GB of data (more than 3 million reviews). Therefore, the queries take quite some time to execute. Plus, all those queries are stored as backup files on the server. As we did not know that feature, we filled up the server capacity of 50GB. We had to proceed to a cleaning of the disk through the command line (see 3.9 Technical Problem). After this issue, we have decided to reduce the size of our database and reduce the scope of our project.

Hence, for the sake of efficiency and optimization we reduced the data to 10 years data (2008-2017) and to five cities in the United States (Cleveland, Charlotte, Las Vegas, Phoenix, Pittsburgh). We have proceeded to some similar cleaning operation, as shown in the Appendix I. Database Cleaning Operations.

3.8 Security Aspects

From security aspects of MySQL, the implementation goes one step further beyond a simple user-password scheme and enables associating each user with a network location to prevent access from unwanted or untrusted locations. When using MySQL Workbench for our project, we made sure to create the account through the Enterprise server and using the specified address "db-vm-

25.el.eee.intern" as it defines which clients can connect to the server and from which machines. In fact, MySQL can use the knowledge of those extra restrictions to prevent authentication attempts for connections coming from unknown locations.

When creating the MySQL server address, the admin created three other users for the other members within the group and assigned them privileges which define what each user can do once they have been authorized to connect. The privileges can be either global, granting specific rights without restrictions, or limited to a specific object such as database or table. In our case, the admin gave the other users privileges to connect, add, modify, or delete tables using queries, as we usually communicate with each other before alternating any tables or database. The feature on Github allows us to track changes made in the code and by which user. However, using the GRANT and REVOKE statements to control access to MySQL is important to not grant more privileges than necessary. The choice of password is critical, as special programs exist to break passwords, and if your computer becomes compromised, the intruder can take the full list of passwords and use them. Using complex passwords with letters and numbers, a mix of upper case and special characters makes it much more challenging to break. The admin used a mix of special characters and numbers to create a strong password for the server. We have also created backups for our databases so that we can recover the data and be up and running again in case problems occur, such as system crashes, hardware failures, or users deleting data by mistake.

There are some security aspects that we would have liked to implement in our project such as adding a user that can only have access to view the database and tables but is not authorized to alter or delete any tables. Ideally, this would be for a customer user to the project. Another feature that can increase security is using certificate-based authentication which simply exists as a combination of the SSL used in connection encryption as well as the privilege system. It enables two-factor authentication where it is no longer sufficient to know username and password, but one also has to provide a valid certificate to be allowed to connect.

3.9 Technical Problem

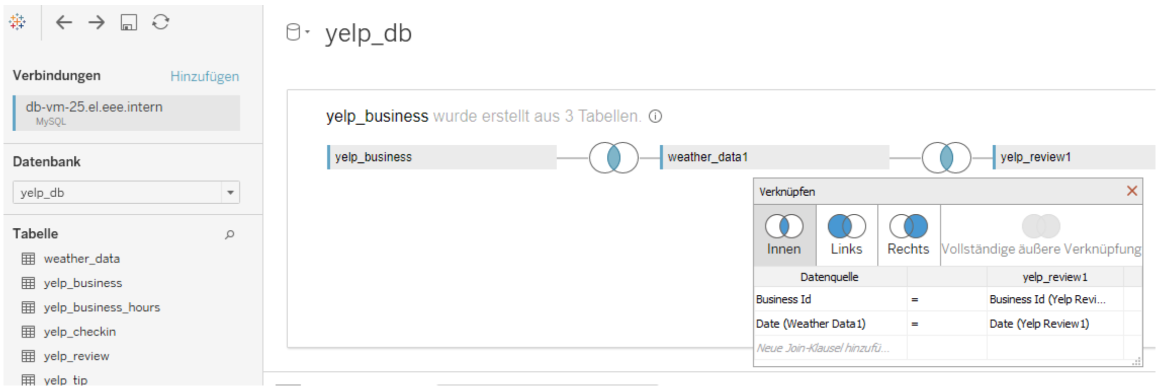
Due to the complexity and size of our data, the loading phase was extremely time consuming, and we faced many failures. Those failures, lead to an overload of the memory capacity of the Virtual Machine and could not access the Server. Due to administrator rights, we could not access the backup logs files of MySQL workbench on the Server (VM). We had to contact EntrepriseLab to increase the memory capacity and erase the log files. It took two to three weeks to fix the server. After this issue, we have decided to reduce the scope of the project (See Appendix I. Database Cleaning operations).

4 Visualization

As to create more sophisticated dashboards with a visual overview of city weather data and the respective star rating, Tableau was applied.

4.1 Database Preparation

After creating the MySQL database on the Virtual Machine with all necessary tables, the database yelp_db had to be connected to Tableau so the data could be accessed directly to create the visualizations. Therefore, a connection to the server address "db-vm-25.el.eee.intern" was established using the dedicated username and password (as described in 3.8 Security Aspects). Once the database could be accessed directly from Tableau, the tables had to be linked in order to create a single table with all necessary columns for the visualizations, making sure that the variables are joined correctly.



yelp_business wurde erstellt aus 3 Tabellen.

Datenquelle		yelp_review1
Business Id	=	Business Id (Yelp Revi...
Date (Weather Data1)	=	Date (Yelp Review1)
Neue Join-Klausel hinzufü...		

Figure 13: Linking tables in Tableau

To achieve this, the tables yelp_business, weather_data1 and yelp_review1 were linked through following keys:

- Business ID
- City
- Date (of weather and of review)

Once the linked table was prepared, the graphical data representation could be approached.

4.2 Design of Dashboards

First, separate work sheets were created with the display of weather and precipitation data and star rating data. The according differences in these variables were visualized through color and object size. Afterwards, the work sheets were combined to create an interactive dashboard.

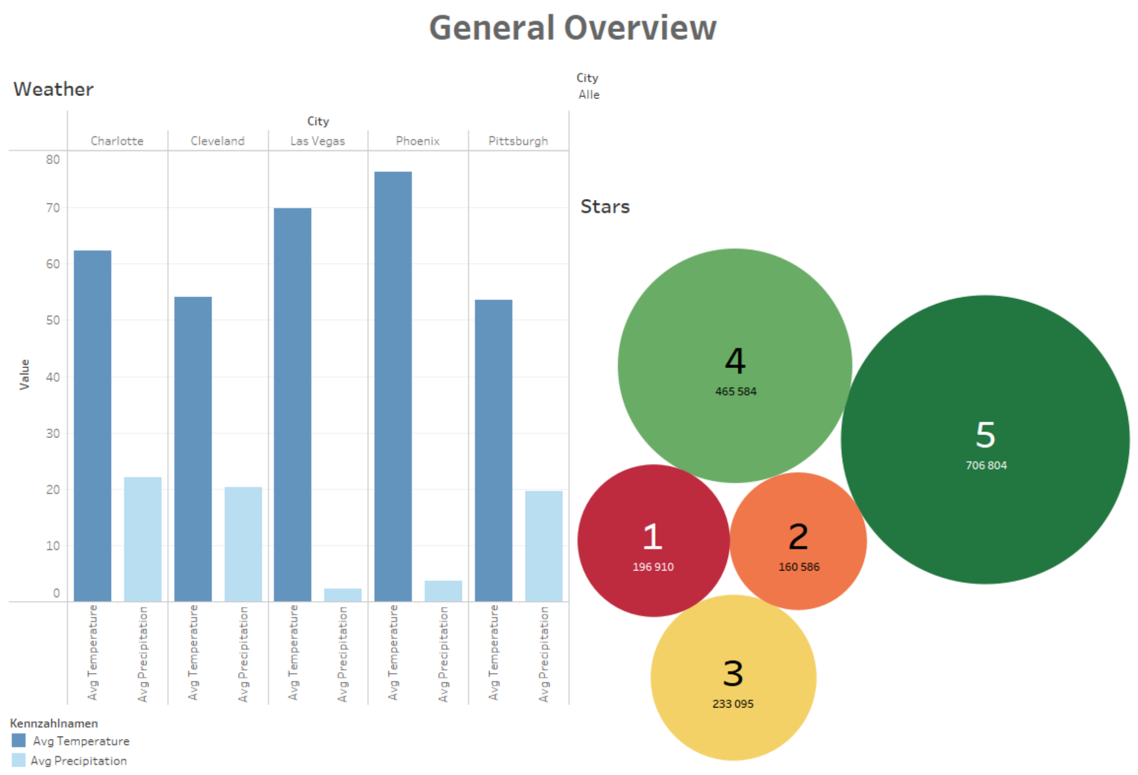


Figure 14: Overview of Avg. Temperature (dark blue) and Avg. Precipitation (light blue) and star rating (circles) in the five US cities

Already from the graphical overview some conclusions could be drawn, especially as to temperature data. It could be clearly determined that the city of Phoenix has the highest average temperature, while Las Vegas has the least precipitation of all cities. As for the rating it could be said that a five-star-rating was given most often and a two-star-rating least often across all cities.

To get a visual understanding of the distribution of the different star ratings across cities, a map was created with the use of longitude and latitude information for all gastronomic businesses. With this, all venues were visually mapped and depicted by a dot in the color which represented their average rating.

In the general view which shows all cities, some outliers can be identified by being mapped to other cities, such the red dot near San Diego.

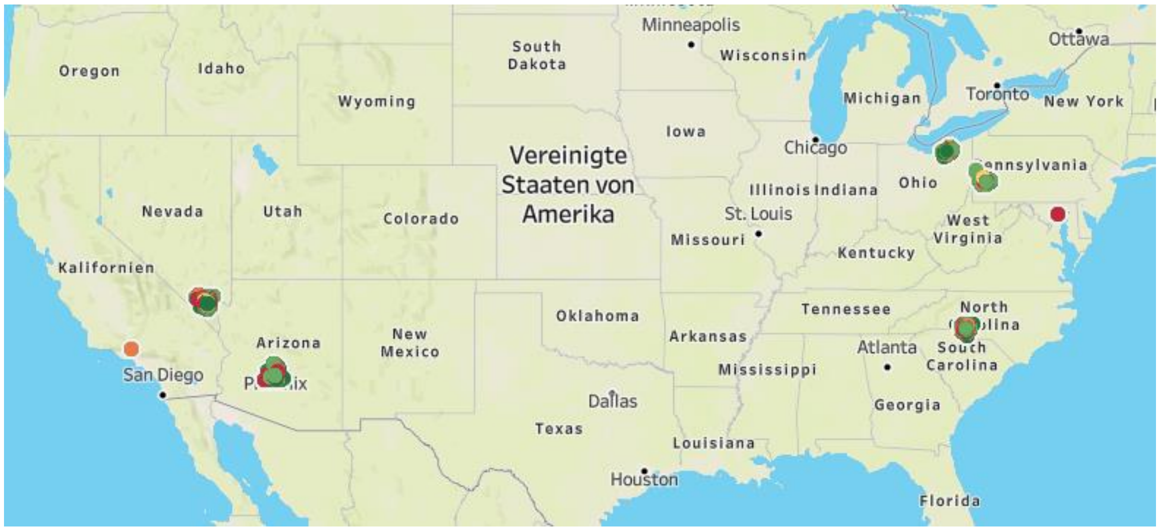


Figure 15: All cities mapped business rating overview

By using the filter and selecting one of the five analysed cities, a clearer picture about the rating distribution of the various gastronomic businesses can be achieved. By hovering over the venue points, more detailed information about the business and its reviews is being displayed.

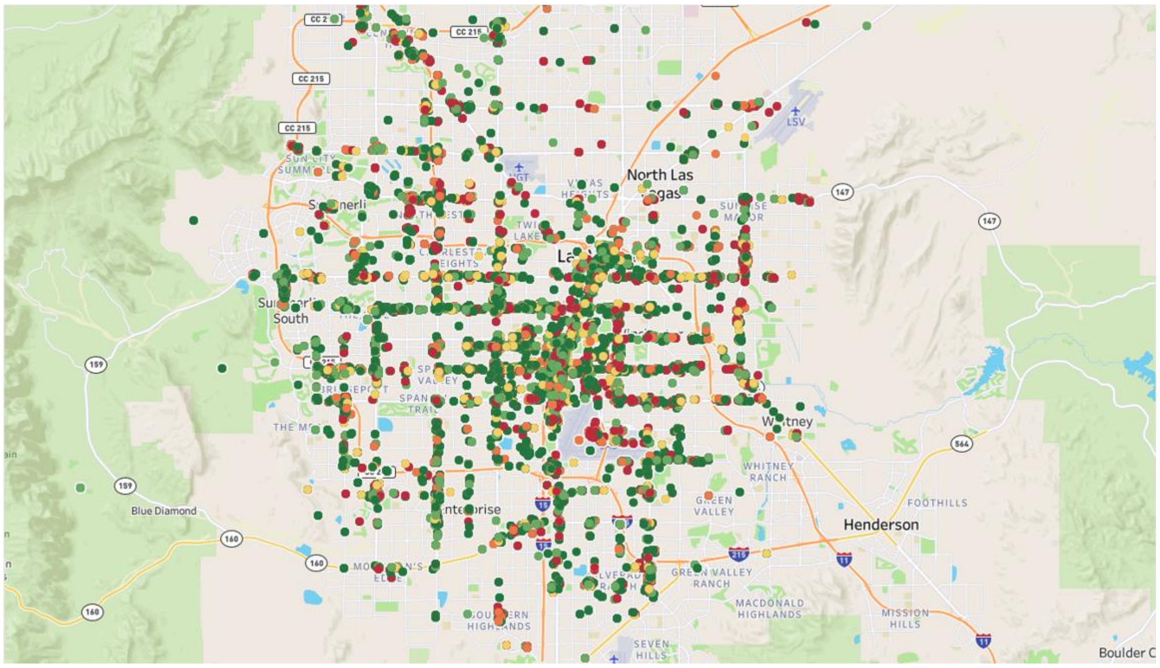


Figure 16: Las Vegas mapped business rating overview

Such a visualization might be of great interest for business owners or investors, who want to get a fast and simple overview of the area with its competitor density or rating.

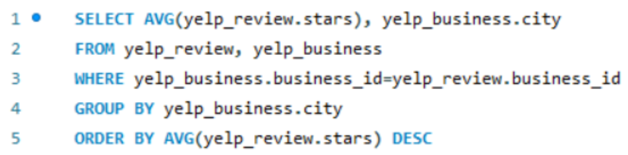
5 Analysis

The database contains data about restaurants, reviews, users, weather data, and many other data. The primary goal of this database is to assist restaurant managers, investors, and other leaders in decision process regarding hospitality service. In the following paragraphs, we will analyse how those decision makers will use our database to lean their decisions.

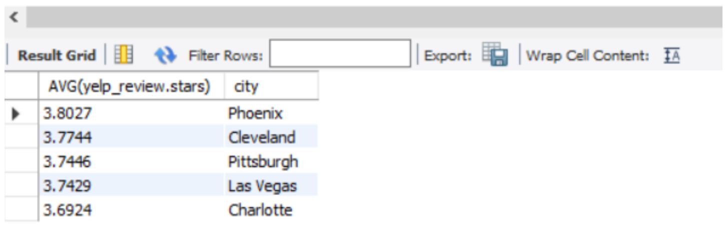
First, we show an analysis whether there is a difference between the selected cities in reviews rating, then we present the impact analysis of weather on restaurant reviews and finally we show some SQL queries that provide an overview of the power of our database and how it can assist restaurant managers and investors.

Analysis 1: Difference between cities

Through SQL request we analysed which city has the highest average star review, which city has the poorest weather and so on. Below are some SQL queries and their results:



```
SELECT AVG(yelp_review.stars), yelp_business.city
FROM yelp_review, yelp_business
WHERE yelp_business.business_id=yelp_review.business_id
GROUP BY yelp_business.city
ORDER BY AVG(yelp_review.stars) DESC
```



AVG(yelp_review.stars)	city
3.8027	Phoenix
3.7744	Cleveland
3.7446	Pittsburgh
3.7429	Las Vegas
3.6924	Charlotte

Figure 17 exe SQL 5

a) Phoenix is the city with the highest average star reviews.

b) Cleveland, Pittsburgh, and Las Vegas seem to have the same average star reviews.

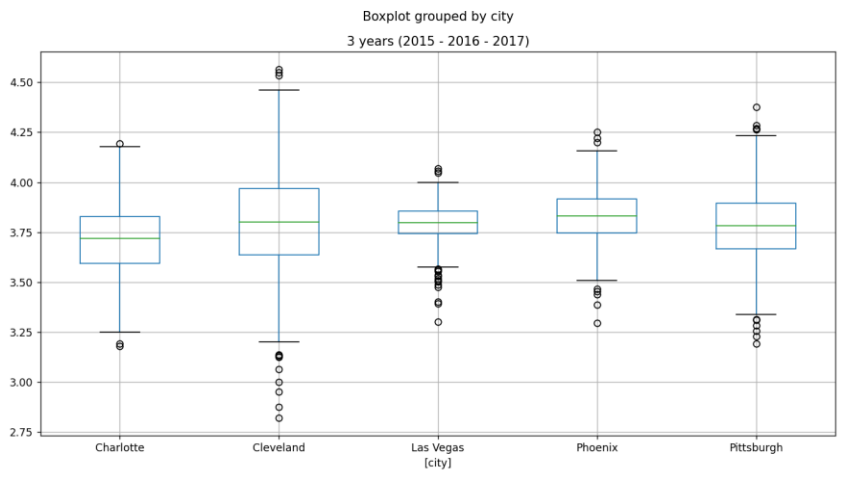
c) Charlotte has the lowest average score.

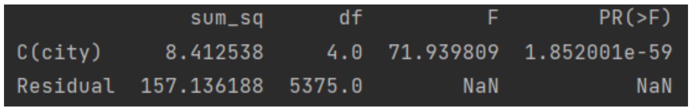
From those findings we can examine the difference between the cities: is there statistical difference in average review?

To answer that question, we will first draw a boxplot and then apply an ANOVA test to have statistical proof. The dataset we will use for this analysis can be found above in the paragraph 4.2 Database Query.

On the above plot, the average star review is shown on the y-axis and the different cities on the x-axis. From this chart we can state that the median of average daily review is equal for every city.

This chart is based on 10 years of data; therefore, the median tends to the same number probably due to same distribution behavior. Hence, we reduced the scope to observe more fluctuations.



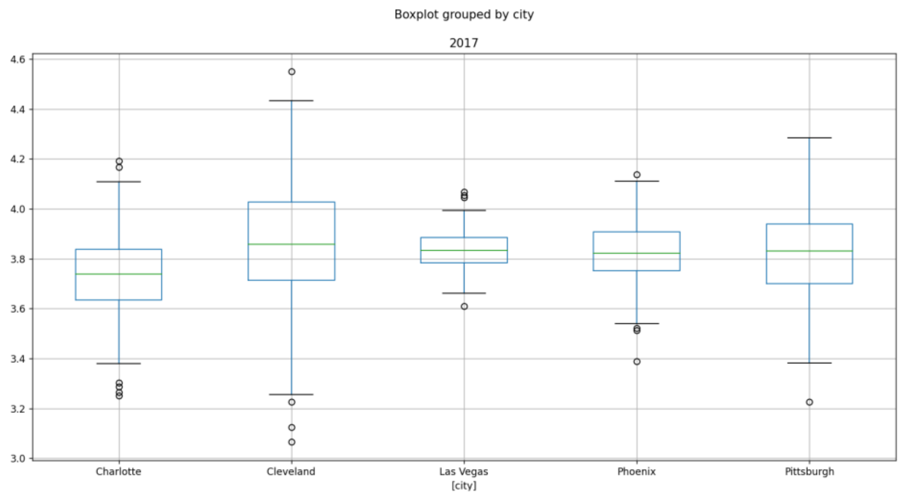


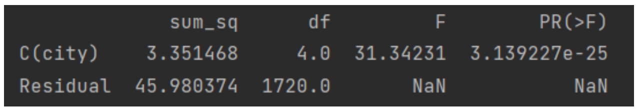
```
              sum_sq      df          F        PR(>F)
C(city)     8.412538     4.0  71.939809  1.852001e-59
Residual  157.136188  5375.0        NaN           NaN
```

Figure 18: Difference Rating 3yr py

Based on the three last years of the data, the city of Charlotte exhibits a median average star review lower than 3.75. Compared to the other cities, Charlotte seems to stand out.

This result proves that there is a statistical difference between those cities based on three years of data. Now we will plot and test the same hypothesis but on one year data from 2017.





```
             sum_sq      df         F        PR(>F)
C(city)    3.351468     4.0  31.34231  3.139227e-25
Residual  45.980374  1720.0       NaN           NaN
```

Figure 19 Difference Rating 2017 py

For the year 2017, the same conclusion applies. There is a statistical difference between those cities. Once again, we suspect Charlotte to be different from the others.

From this first analysis, we can state that there is a difference of review rating between those five cities in the USA. Therefore, a restaurant manager, thanks to our database, can be aware of this difference and, if he is considering opening a restaurant in Charlotte, he knows, the reviews are lower in this city. Perhaps, the customers in this city are more sensitive about hospitality service or maybe an external factor, like bad weather, high crime rate, impact the customers experience.

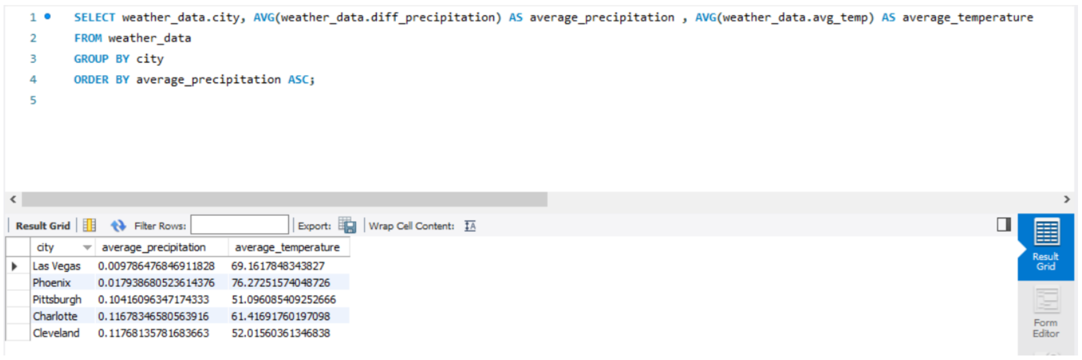
Analysis 2: Weather impact on customer experience

We assume that, if Charlotte has worse reviews than the four other cities, it is probably due to external factors (people's mindset, weather, crime rate, etc.). In our opinion, there is no particular reason why the hospitality service is poorer in this city (one possibility is the number of reviews, consult the Appendix II. SQL request, to see the difference in number of reviews per city).

Therefore, in this section, we will analyse the influence of one external factor, weather, on restaurant reviews. Our hypothesis is that a customer will give a lower review if he experiences bad weather (heavy rainfall, extreme temperature). The weather impact can be direct (conscious) or indirect (unconscious). If the customer is having a coffee on a terrasse and it starts to rain, then, if the restaurants has not the appropriate equipment for rain, the client will have a bad experience and leave a bad review, which is due to a direct impact of weather on customer experience. This assumption is supported by many studies which show that weather has an impact on people's mood. So, if a customer is having a coffee inside and it is raining, the rain can affect the customer's mood and he/she might not enjoy his/her coffee as if it was sunny. Our model cannot measure such psychological bias, but it gives an idea of how weather, as an external factor, can affect a customer's experience.

To analyse the impact of weather on client satisfaction, we will use the previous conclusion that Charlotte has a lower review rating than the other cities. We will compare Charlotte to Las Vegas due to their difference in customer reviews and weather conditions.

We start by a simple SQL observation:



```sql
SELECT weather_data.city, AVG(weather_data.diff_precipitation) AS average_precipitation , AVG(weather_data.avg_temp) AS average_temperature
FROM weather_data
GROUP BY city
ORDER BY average_precipitation ASC;
```

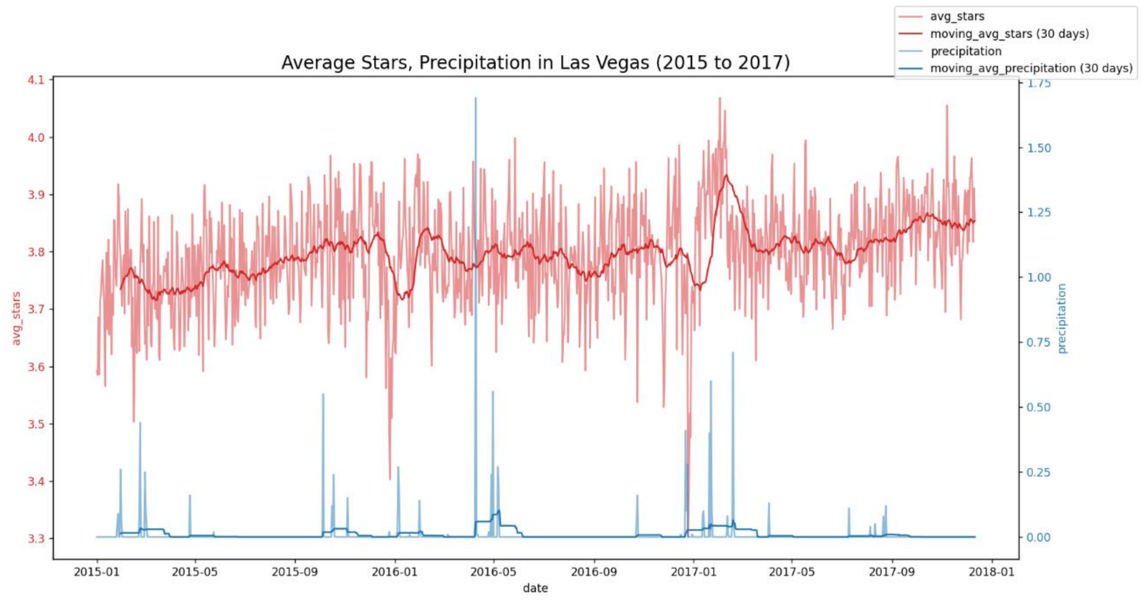
city	average_precipitation	average_temperature
Las Vegas	0.009786476846911828	69.1617848343827
Phoenix	0.017938680523614376	76.27251574048726
Pittsburgh	0.10416096347174333	51.096085409252666
Charlotte	0.11678346580563916	61.41691760197098
Cleveland	0.11768135781683663	52.01560361346838

Figure 20 exe SQL 6

a) Las Vegas is the city with least precipitation, then comes Phoenix. Although Phoenix has the highest average temperature, extremely high temperature is not enjoyable.

b) Cleveland has the poorest weather with low temperatures and lots of rain, then comes Charlotte.

Now we want to analyse the weather and the ratings behavior of the two cities, to see if there are any visual patterns that will indicate weather impact on review ratings.



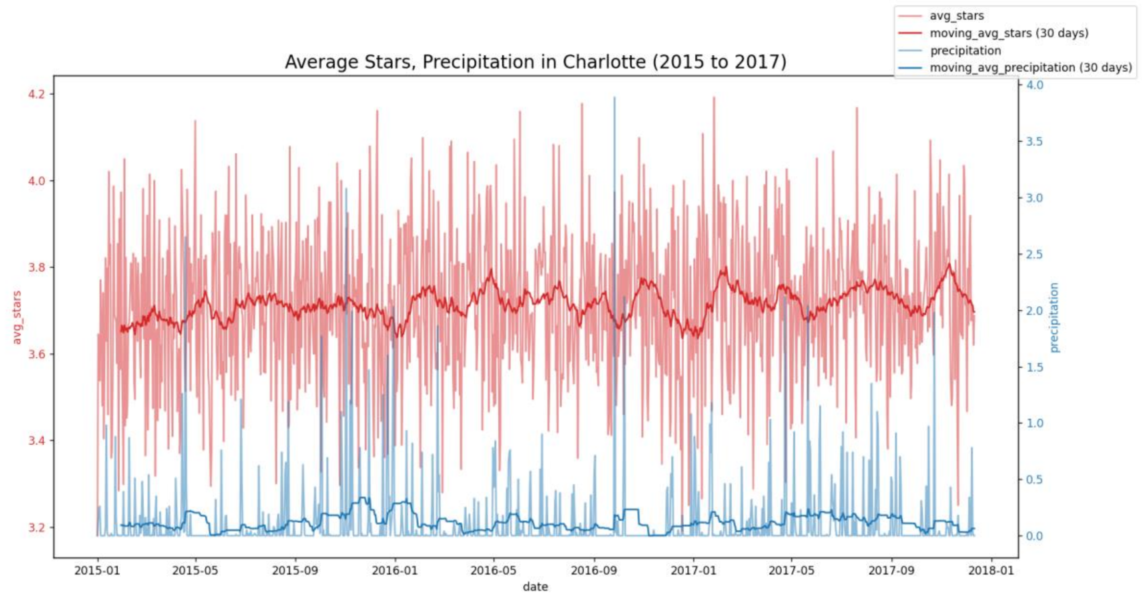
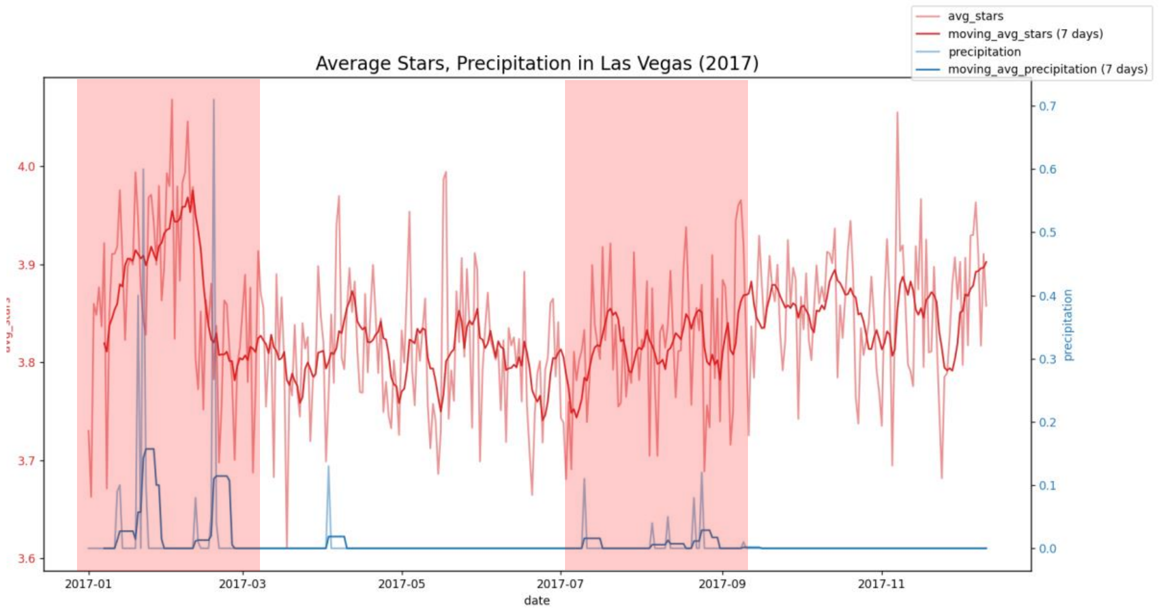


Figure 21 Weather and Rating Behavior on 3 year data in Las Vegas and Charlotte

As can be seen on the above plots, the weather distribution is clearly different. There are really small amounts of rain in Las Vegas. On the other hand, Charlotte seems to have rain at least every month (30 days moving average). Regarding the ratings distributions, it seems like Las Vegas has ratings between 3.7 and 3.8 on monthly basis (30 days moving average). Charlotte seems to be around 3.6 and 3.7.

However, it is hard to distinguish any patterns to conclude any link between weather and review ratings. Therefore, we plot the data for the last year, 2017.



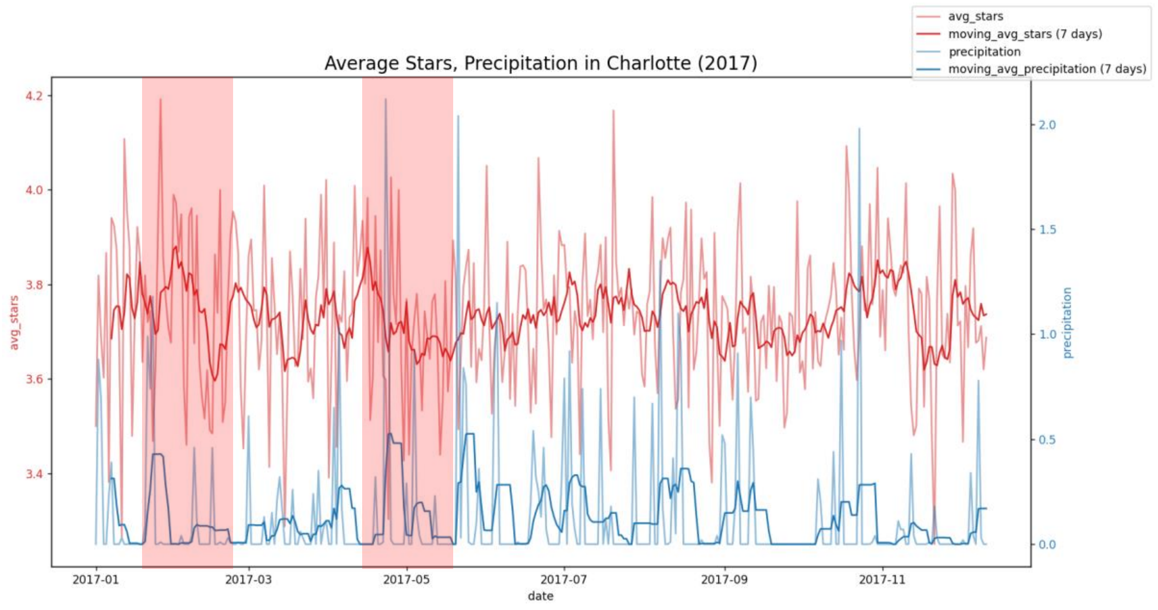
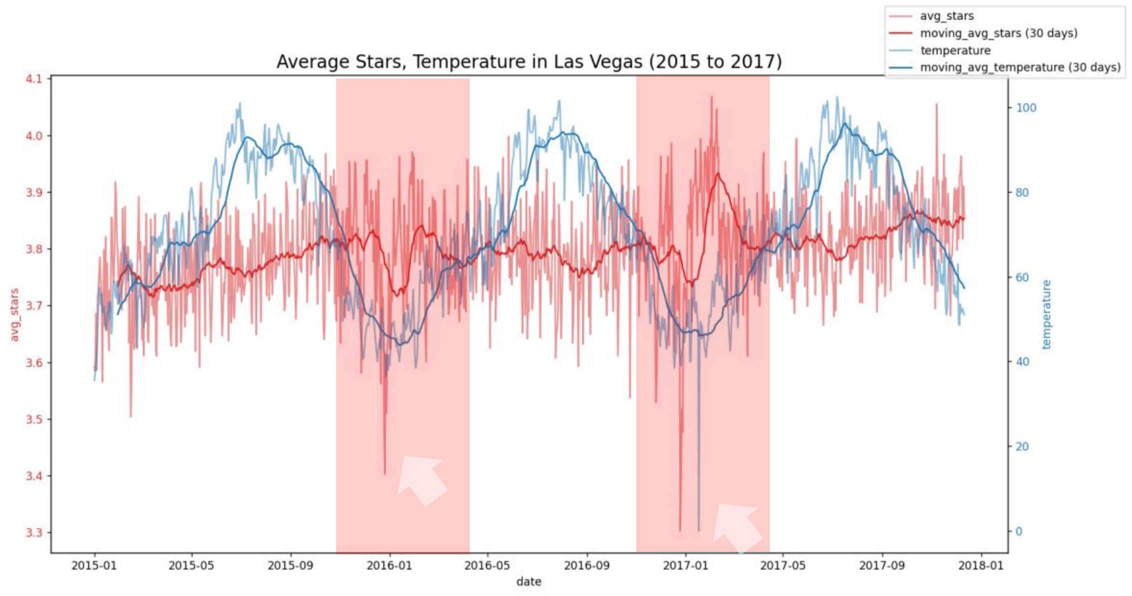


Figure 22 Weather and Rating Behavior 2017 in Las Vegas and Charlotte

There are no clear visual patterns indicating a link between star ratings and precipitation. Therefore, we will build a correlation matrix and a linear regression to understand the impact of weather on review ratings. Now we will observe the average temperature and the ratings for Charlotte and Las Vegas in 2013 - 2017 (See the Appendix III. Complement to Analysis, to visualize ratings and temperature 2017).



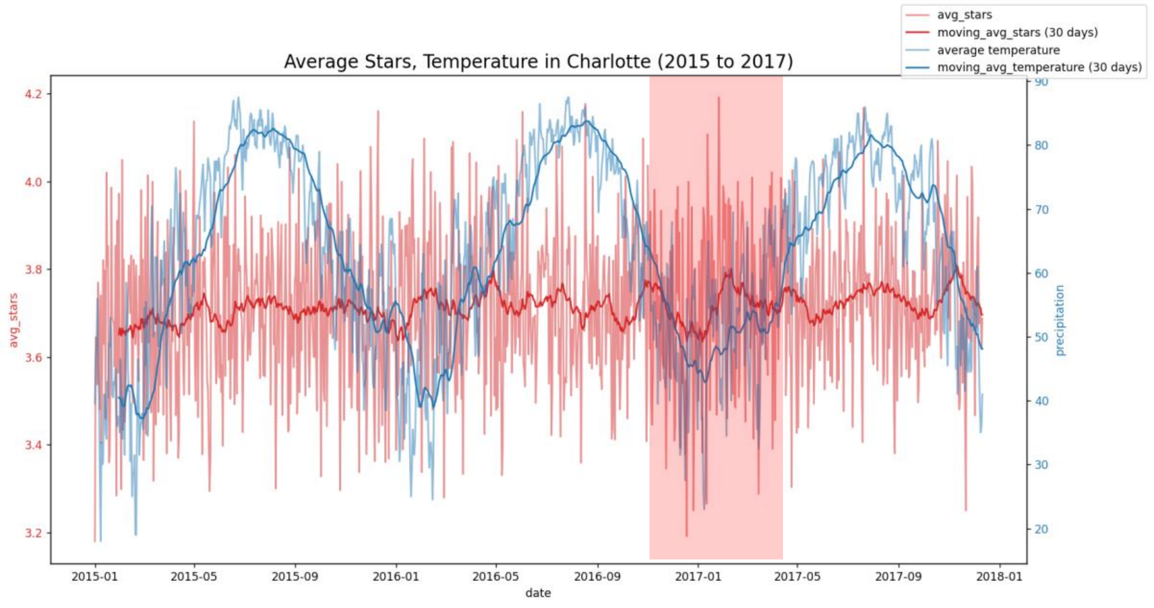
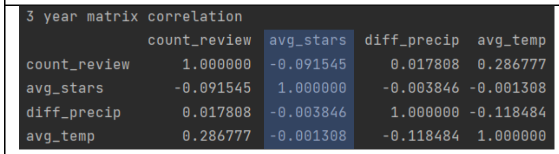


Figure 23 Temperature and Rating Behavior on 3 years data in Las Vegas and Charlotte

In Las Vegas, we can guess a pattern during winter season (around January), but this effect contradicts itself. However, we can still observe a massive drop in January both in temperature (blue) and ratings (red). On the other hand, it is harder to observe such patterns in Charlotte.

Therefore, we will observe correlation matrix and a linear regression over three years and one year to understand the effect of temperature and rain on rating reviews.

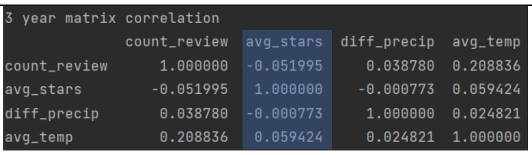
Las Vegas



```
3 year matrix correlation
              count_review  avg_stars  diff_precip  avg_temp
count_review      1.000000  -0.091545     0.017808  0.286777
avg_stars        -0.091545   1.000000    -0.003846 -0.001308
diff_precip       0.017808  -0.003846     1.000000 -0.118484
avg_temp          0.286777  -0.001308    -0.118484  1.000000
```

No correlation

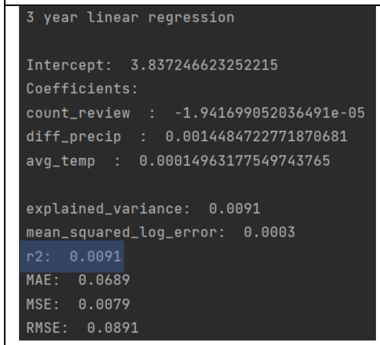
Charlotte



```
3 year matrix correlation
              count_review  avg_stars  diff_precip  avg_temp
count_review      1.000000  -0.051995     0.038780  0.208836
avg_stars        -0.051995   1.000000    -0.000773  0.059424
diff_precip       0.038780  -0.000773     1.000000  0.024821
avg_temp          0.208836   0.059424     0.024821  1.000000
```

No correlation

Las Vegas

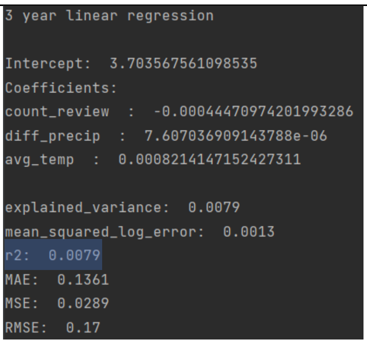


```
3 year linear regression

Intercept:  3.837246623252215
Coefficients:
count_review  :  -1.941699052036491e-05
diff_precip  :  0.0014484722771870681
avg_temp  :  0.00014963177549743765

explained_variance:  0.0091
mean_squared_log_error:  0.0003
r2:  0.0091
MAE:  0.0689
MSE:  0.0079
RMSE:  0.0891
```

Low R-squared and coefficient equal to 0

→ No significance

Charlotte

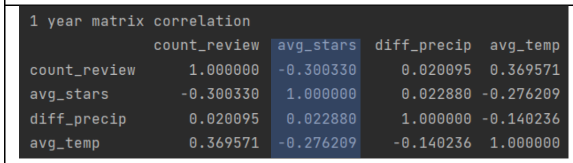


```
3 year linear regression

Intercept:  3.703567561098535
Coefficients:
count_review  :  -0.00044470974201993286
diff_precip  :  7.607036909143788e-06
avg_temp  :  0.0008214147152427311

explained_variance:  0.0079
mean_squared_log_error:  0.0013
r2:  0.0079
MAE:  0.1361
MSE:  0.0289
RMSE:  0.17
```

Low R-squared and coefficient equal to 0

→ No significance

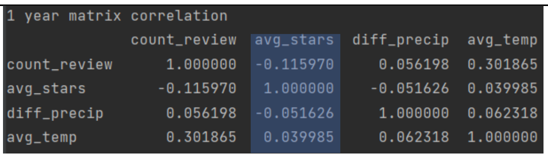
Las Vegas



```
1 year matrix correlation
              count_review  avg_stars  diff_precip  avg_temp
count_review      1.000000  -0.300330     0.020095  0.369571
avg_stars        -0.300330   1.000000     0.022880 -0.276209
diff_precip       0.020095   0.022880     1.000000 -0.140236
avg_temp          0.369571  -0.276209    -0.140236  1.000000
```

No correlation (except for temperature -0.3)

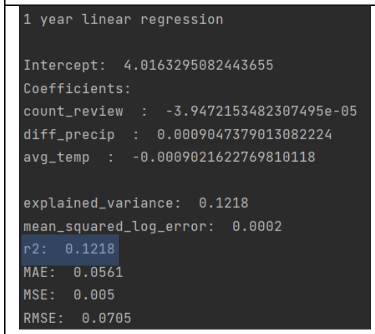
Charlotte



```
1 year matrix correlation
              count_review  avg_stars  diff_precip  avg_temp
count_review      1.000000  -0.115970     0.056198  0.301865
avg_stars        -0.115970   1.000000    -0.051626  0.039985
diff_precip       0.056198  -0.051626     1.000000  0.062318
avg_temp          0.301865   0.039985     0.062318  1.000000
```

No correlation

Las Vegas

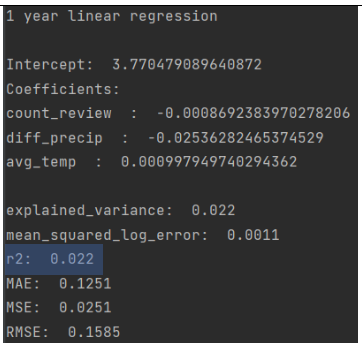


```
1 year linear regression

Intercept:  4.0163295082443655
Coefficients:
count_review  :  -3.9472153482307495e-05
diff_precip  :  0.0009047379013082224
avg_temp  :  -0.0009021622769810118

explained_variance:  0.1218
mean_squared_log_error:  0.0002
r2:  0.1218
MAE:  0.0561
MSE:  0.005
RMSE:  0.0705
```

12% R-squared and coefficient equal to 0

→ No significance

Charlotte



```
1 year linear regression

Intercept:  3.770479089640872
Coefficients:
count_review  :  -0.0008692383970278206
diff_precip  :  -0.02536282465374529
avg_temp  :  0.000997949740294362

explained_variance:  0.022
mean_squared_log_error:  0.0011
r2:  0.022
MAE:  0.1251
MSE:  0.0251
RMSE:  0.1585
```

Low R-squared and coefficient equal to 0

→ No significance

Figure 24 Summary Analysis

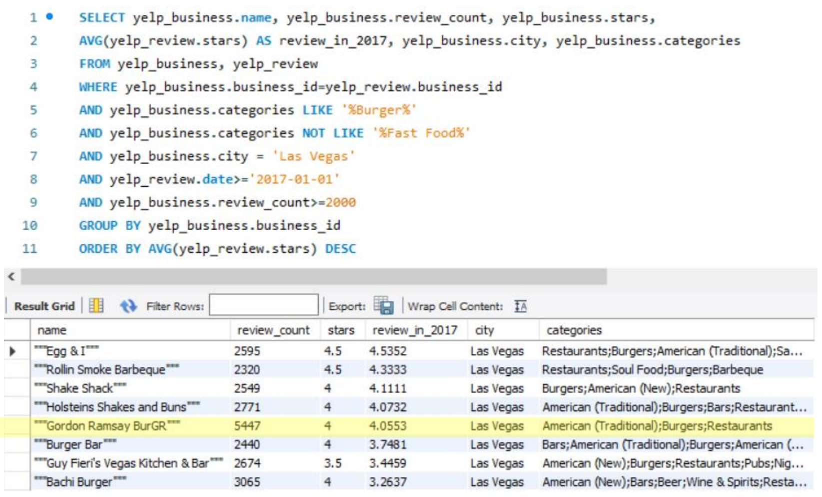
As can be seen, the results are non-significant. Therefore, based on statistical evidence, we conclude that weather has no impact on reviews in these cities at these periods. To pick up on the study[3] mention in the use case, the study was made on 32 restaurants in Florida, therefore the scope is different and so are the result.

The restaurant manager or the investor willing to open a restaurant in Las Vegas or in Charlotte now knows that the weather has no impact on the customer experience. Thanks to our database and our analysis, leaders know that weather is not a concerning external factor. However, they can still consult our database to observe weather in the cities where they are considering opening a new restaurant, to plan on buying the right equipment to improve customer's experience and outperform other businesses.

Analysis 3: Information source and comparing tool

This database contains information about restaurants, restaurants reviews and external factors like weather data, but it also contains valuable information about opening hours, yelp users, tips and more. In this last section we will study several SQL requests to see how investors and restaurant managers can take advantage of the database to build a strategy for their restaurants.

We will use an example to focus this analysis. Let say the famous British chef, Gordon Ramsay, wants to have his restaurant, *'Gordon Ramsay BurGR'* to be the top 1 burger restaurant in Las Vegas (ranking based 2017 reviews and on restaurants with more than 2000 reviews that are not Fast Food).



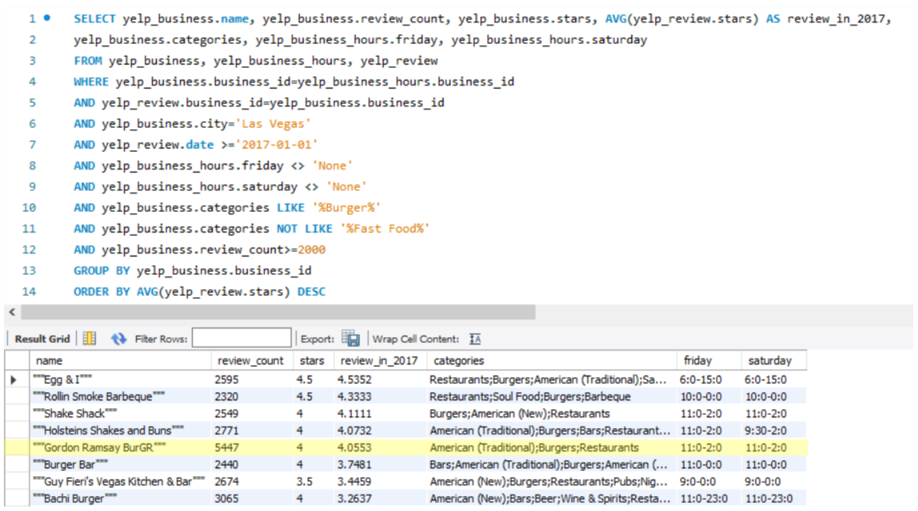
```sql
SELECT yelp_business.name, yelp_business.review_count, yelp_business.stars,
AVG(yelp_review.stars) AS review_in_2017, yelp_business.city, yelp_business.categories
FROM yelp_business, yelp_review
WHERE yelp_business.business_id=yelp_review.business_id
AND yelp_business.categories LIKE '%Burger%'
AND yelp_business.categories NOT LIKE '%Fast Food%'
AND yelp_business.city = 'Las Vegas'
AND yelp_review.date>='2017-01-01'
AND yelp_business.review_count>=2000
GROUP BY yelp_business.business_id
ORDER BY AVG(yelp_review.stars) DESC
```

name	review_count	stars	review_in_2017	city	categories
"""Egg & I"""	2595	4.5	4.5352	Las Vegas	Restaurants;Burgers;American (Traditional);Sa...
"""Rollin Smoke Barbeque"""	2320	4.5	4.3333	Las Vegas	Restaurants;Soul Food;Burgers;Barbeque
"""Shake Shack"""	2549	4	4.1111	Las Vegas	Burgers;American (New);Restaurants
"""Holsteins Shakes and Buns"""	2771	4	4.0732	Las Vegas	American (Traditional);Burgers;Bars;Restaurant...
"""Gordon Ramsay BurGR"""	5447	4	4.0553	Las Vegas	American (Traditional);Burgers;Restaurants
"""Burger Bar"""	2440	4	3.7481	Las Vegas	Bars;American (Traditional);Burgers;American (...
"""Guy Fieri's Vegas Kitchen & Bar"""	2674	3.5	3.4459	Las Vegas	American (New);Burgers;Restaurants;Pubs;Nig...
"""Bachi Burger"""	3065	4	3.2637	Las Vegas	American (New);Bars;Beer;Wine & Spirits;Resta...

Figure 25 exe SQL 7

As his restaurant is operating in Las Vegas, famous for the nightlife, he is wondering if his competitors are operating as late as his him.

[3] https://www.sciencedaily.com/releases/2019/04/190422082243.htm

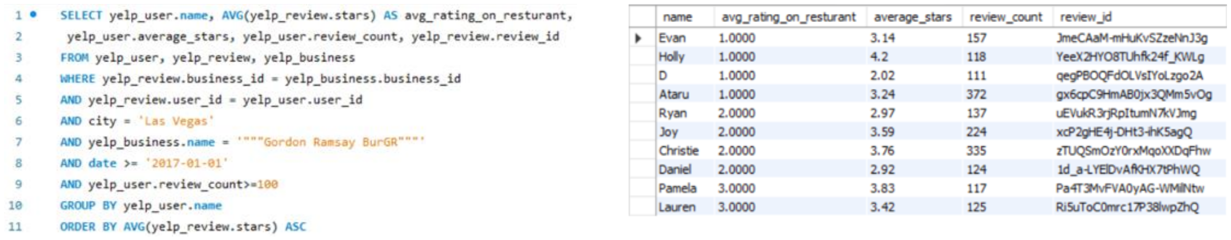


```sql
SELECT yelp_business.name, yelp_business.review_count, yelp_business.stars, AVG(yelp_review.stars) AS review_in_2017,
yelp_business.categories, yelp_business_hours.friday, yelp_business_hours.saturday
FROM yelp_business, yelp_business_hours, yelp_review
WHERE yelp_business.business_id=yelp_business_hours.business_id
AND yelp_review.business_id=yelp_business.business_id
AND yelp_business.city='Las Vegas'
AND yelp_review.date >='2017-01-01'
AND yelp_business_hours.friday <> 'None'
AND yelp_business_hours.saturday <> 'None'
AND yelp_business.categories LIKE '%Burger%'
AND yelp_business.categories NOT LIKE '%Fast Food%'
AND yelp_business.review_count>=2000
GROUP BY yelp_business.business_id
ORDER BY AVG(yelp_review.stars) DESC
```

name	review_count	stars	review_in_2017	categories	friday	saturday
"""Egg & I"""	2595	4.5	4.5352	Restaurants;Burgers;American (Traditional);Sa...	6:0-15:0	6:0-15:0
"""Rollin Smoke Barbeque"""	2320	4.5	4.3333	Restaurants;Soul Food;Burgers;Barbeque	10:0-0:0	10:0-0:0
"""Shake Shack"""	2549	4	4.1111	Burgers;American (New);Restaurants	11:0-2:0	11:0-2:0
"""Holsteins Shakes and Buns"""	2771	4	4.0732	American (Traditional);Burgers;Bars;Restaurant...	11:0-2:0	9:30-2:0
"""Gordon Ramsay BurGR"""	5447	4	4.0553	American (Traditional);Burgers;Restaurants	11:0-2:0	11:0-2:0
"""Burger Bar"""	2440	4	3.7481	Bars;American (Traditional);Burgers;American (...	11:0-0:0	11:0-0:0
"""Guy Fieri's Vegas Kitchen & Bar"""	2674	3.5	3.4459	American (New);Burgers;Restaurants;Pubs;Nig...	9:0-0:0	9:0-0:0
"""Bachi Burger"""	3065	4	3.2637	American (New);Bars;Beer;Wine & Spirits;Resta...	11:0-23:0	11:0-23:0

Figure 26 exe SQL 8

As we can see his two direct rivals are operating the same hours as his restaurant. It seems like he cannot differentiate his business on this strategy. Therefore, he wants to analyse the reviews that sunk his star ratings that year.



```sql
SELECT yelp_user.name, AVG(yelp_review.stars) AS avg_rating_on_resturant,
 yelp_user.average_stars, yelp_user.review_count, yelp_review.review_id
FROM yelp_user, yelp_review, yelp_business
WHERE yelp_review.business_id = yelp_business.business_id
AND yelp_review.user_id = yelp_user.user_id
AND city = 'Las Vegas'
AND yelp_business.name = '"""Gordon Ramsay BurGR"""'
AND date >= '2017-01-01'
AND yelp_user.review_count>=100
GROUP BY yelp_user.name
ORDER BY AVG(yelp_review.stars) ASC
```

name	avg_rating_on_resturant	average_stars	review_count	review_id
Evan	1.0000	3.14	157	JmeCAaM-mHuKvSZzeNnJ3g
Holly	1.0000	4.2	118	YeeX2HYO8TUhfk24f_KWLg
D	1.0000	2.02	111	qegPBOQFdOLVsIYoLzgo2A
Ataru	1.0000	3.24	372	gx6cpC9HmAB0jx3QMm5vOg
Ryan	2.0000	2.97	137	uEVukR3rjRpItumN7kVJmg
Joy	2.0000	3.59	224	xcP2gHE4j-DHt3-ihK5agQ
Christie	2.0000	3.76	335	zTUQSmOzY0rxMqoXXDqFhw
Daniel	2.0000	2.92	124	1d_a-LYElDvAfKHX7tPhWQ
Pamela	3.0000	3.83	117	Pa4T3MvFVA0yAG-WMilNtw
Lauren	3.0000	3.42	125	Ri5uToC0mrc17P38lwpZhQ

Figure 27 exe SQL 9

For veracity purposes, we analyse user with a minimum of 100 reviews. We can observe that some users were harsh on Gordon. He can read their comments with the review_id key and improve (see the Appendix). On the other hand, we see the user 'D' made 111 reviews and has an average review of 2.02, therefore this user gives low reviews to business. Plus, this name is suspicious, so Gordon can ask Yelp® to check the authenticity of this user and ask to remove the comment.

To conclude, we have seen the different information the database contains and how restaurants managers can use it as a source of information or as comparison tool. The database is extremely rich, diversified and contains valuable information.

5.1 Decision Support

From the first part of this analysis, we have shown that there is a statistical difference between five cities in the US. The city Charlotte, in North Carolina, has the lowest average rating review compared to Las Vegas, Phoenix, Pittsburgh, and Cleveland. Therefore, our recommended strategy for restaurant managers or investors is to be careful regarding the marking strategy. A marketing strategy based on reviews will suit to cities like Las Vegas, Phoenix, Pittsburgh, and Cleveland but not in Charlotte.

From the second part of this analysis, we know that weather condition has no significant impact on rating review. Sunny and warm weather will not increase customer satisfaction, nor rainy and cold weather will have an impact on the reviews. Hence, our recommended strategy to restaurant managers and investors is still to look at the weather model to adapt their restaurant equipment (umbrella, fan, heater, etc.) and use weather fluctuations to adapt marketing and product strategy (ex. Spiced Latte from Starbucks for winter season). In conclusion, weather conditions will not directly affect customer's experience.

From the last part, we have seen that a manager such as Gordon Ramsay can use the database to obtain information and compare his business to its competitors. Our recommendation to his restaurants is to carefully observe low review ratings, see if he can improve weak points and especially observe if the users are authenticated. His opening hours look appropriate towards compactivity and the environment.

To conclude this analysis, this database is an important source of information for businesses in the United States. Investors and restaurant managers can compare their businesses and throw scientific analysis to improve their strategy. The undeniable strength of this database is its size.

The Team, a passionate Data Scientist team from the HSLU, is happy to offer yelp_db, the new database for gastronomic businesses with 1.8 million reviews, 10 years data, 21'000 restaurants, 1.3 million users, 18'000 weather entries and five cities in the USA.

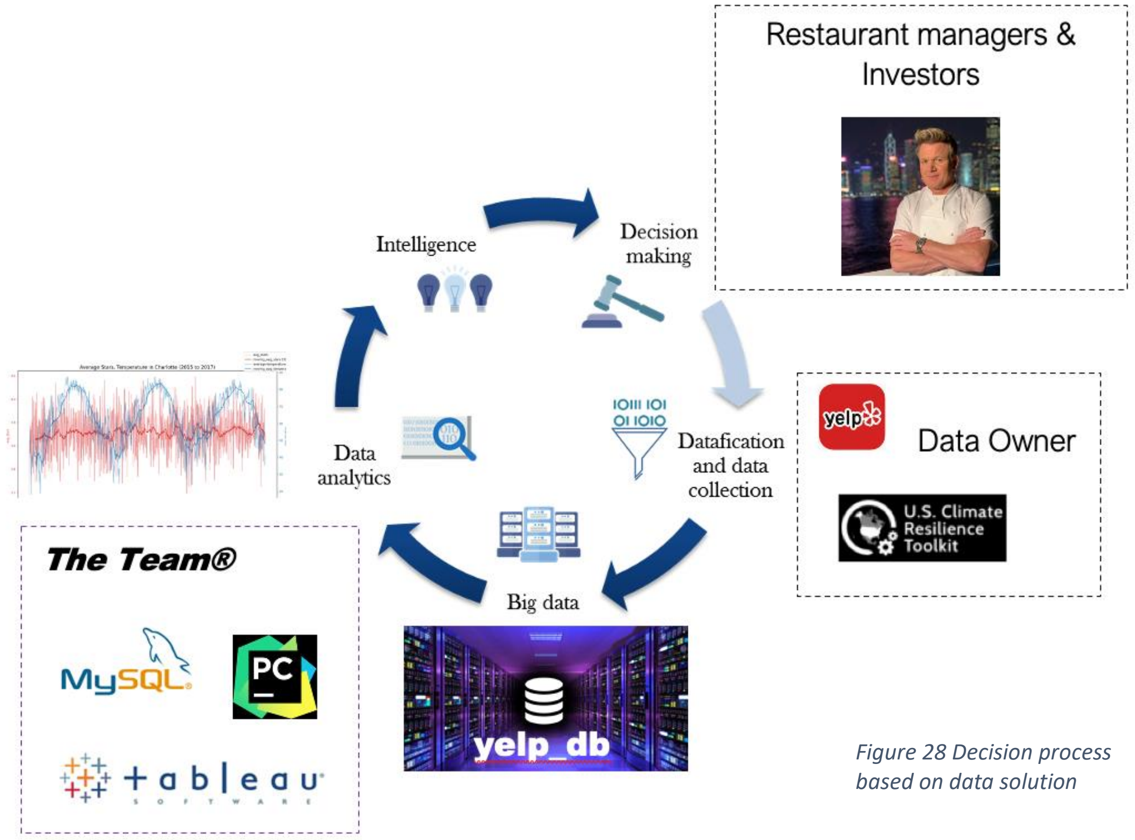


Figure 28 Decision process based on data solution

6 Lessons Learned

After spending a semester developing different project in different domains of data science like Machine Learning, Visualization, Processing and Analysis, we understand how crucial, and complex a database can be. Now, we have built our own on MySQL Server, and have faced many challenges, loading data failure, memory capacity overload, database connection lost, and many other. We experienced how difficult is it to set up a database. Despite the many difficulties we faced, we also had the chance to discover how useful and convenient it was to work with a structure database on MySQL Server.

First, it gives a considerable collaboration advantage. We were all able to access the database and make some queries. One student built the database and ensured the database was running correctly, another student was expert in SQL query and produced rich tables, another one oversaw exporting and exploiting those queries in PyCharm for meaningful analysis, and finally the last student was in charge of connecting Tableau and MySQL workbench to build valuable visualizations.

Secondly, building a database is a long and extensive work. The nice and well-structured SQL tables, visualization and the analysis are just the visible part of the iceberg. The hidden part is all the backend work we performed. Ensuring the server ran on all computers, fixing memory capacity issues, loading data in tables with correct data type, speed up SQL queries, cleaning dataset and database, connecting the SQL Server to Tableau and PyCharm, to quote some.

Finally, with SQL databases it is possible to build an infinite number of data sets with clean and structured data. We were able to create different data sets, in order to analyse different research questions (difference between cities, weather impact) and for Tableau Software, everything from the same data source. We concluded that databases are the pillars of data science.

Figure 29 iceberg analogy

7 Conclusion

The main focus of our project was to examine how weather conditions influence the ratings of gastronomic businesses in the USA and, by applying various data science methods, we were able to come to the conclusion that weather has no significant effect on business reviews. Therefore, our hypothesis, that external factors might influence customer experience, can be rejected in the case of weather and climate.

Nonetheless, the database and visualizations which we have created in this project remain to serve a greater purpose as a helpful tool for restaurateurs and investors to perform various market, location, rating, or competitor analyses.

7.1 Further Analysis

As the underlying hypothesis states, it is argued that various external factors might affect customer experience. Since we have examined only the influence of weather in this project, the analysis of the effects of other external factors on client satisfaction, and thus business ratings, remains to be performed. Some examples of other external factors could be:

- Crime rate in the area around/in the city of the gastronomic business
- Proximity to city center
- Accessibility to public transport

Further investigation of such possible relations would aid entrepreneurs and gastronomic business owners to improve the success rate of their businesses and help to efficiently allocate their investments.

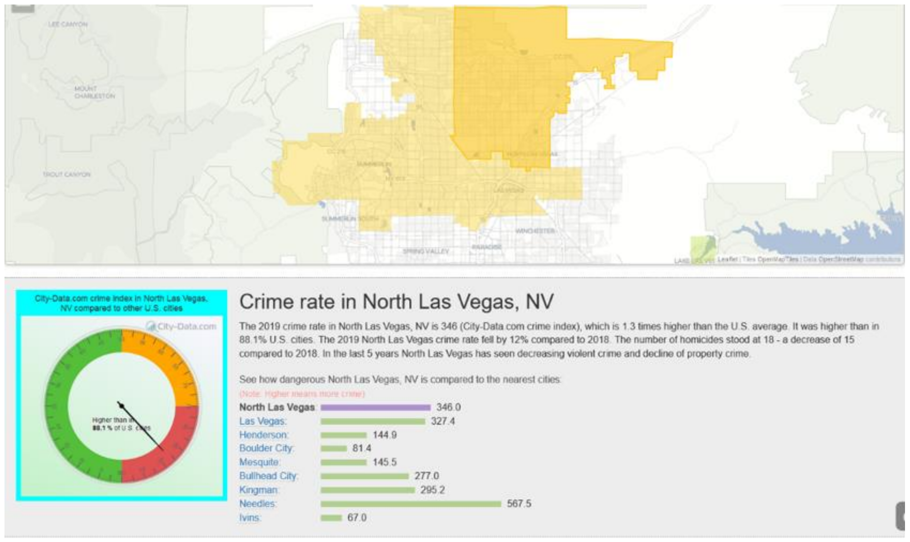
7.1.1 External Factor: Crime Rate

A strong suggestion would be to examine the influence of crime rate in a certain area on the gastronomic business ratings in that specific area. There are indications that lead to the assumption that a possible significant effect could be identified in this regard, as can be seen in the example of North Las Vegas, Nevada.

The 2019 crime rate in the area of North Las Vegas showed a very high number, namely 1.3 timer higher that the US average and thus higher than in 88.1% of US cities.[4] At the same time, from our Tableau visualization of gastronomic business ratings in Las Vegas, we had the impression that North Las Vegas had an increased number of businesses with low customer ratings, as depicted by the red and orange dots in the circled area from the figure below.

[4] https://www.city-data.com/crime/crime-North-Las-Vegas-Nevada.html

We therefore believe that examining a possible relationship between crime rate as an external influence factor and gastronomic business rating might be an interesting starting point for further analysis.


City-Data.com crime index in North Las Vegas, NV compared to other U.S. cities
City-Data.com
Higher than in 88.1 % of U.S. cities

Crime rate in North Las Vegas, NV

The 2019 crime rate in North Las Vegas, NV is 346 (City-Data.com crime index), which is 1.3 times higher than the U.S. average. It was higher than in 88.1% U.S. cities. The 2019 North Las Vegas crime rate fell by 12% compared to 2018. The number of homicides stood at 18 - a decrease of 15 compared to 2018. In the last 5 years North Las Vegas has seen decreasing violent crime and decline of property crime.

See how dangerous North Las Vegas, NV is compared to the nearest cities:
(Note: Higher means more crime)

North Las Vegas	346.0
Las Vegas:	327.4
Henderson:	144.9
Boulder City:	81.4
Mesquite:	145.5
Bullhead City:	277.0
Kingman:	295.2
Needles:	567.5
Ivins:	67.0

Figure 280: Crime rate in North Las Vegas in 2019
Source: https://www.city-data.com/crime/crime-North-Las-Vegas-Nevada.html

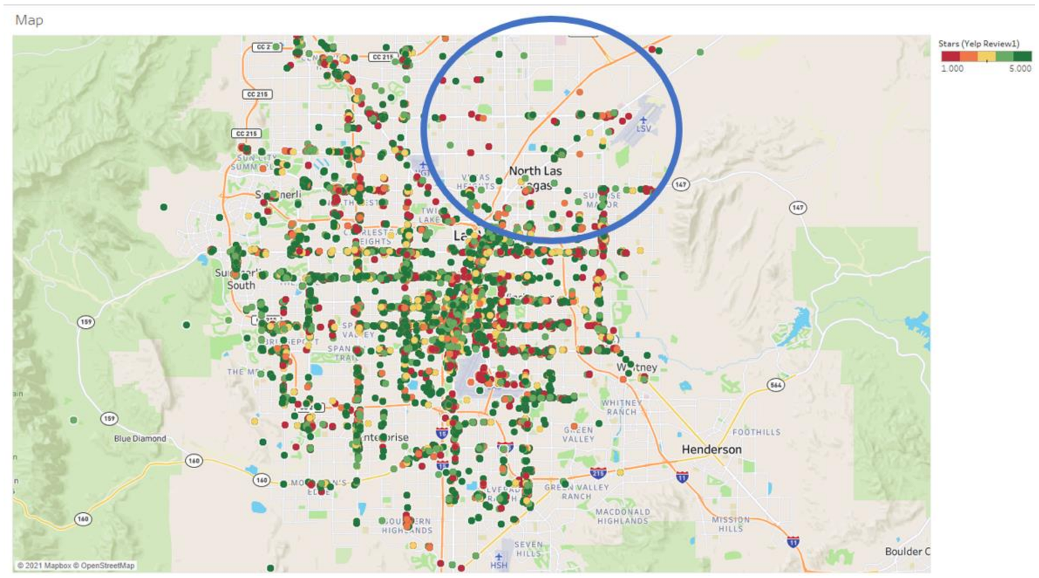
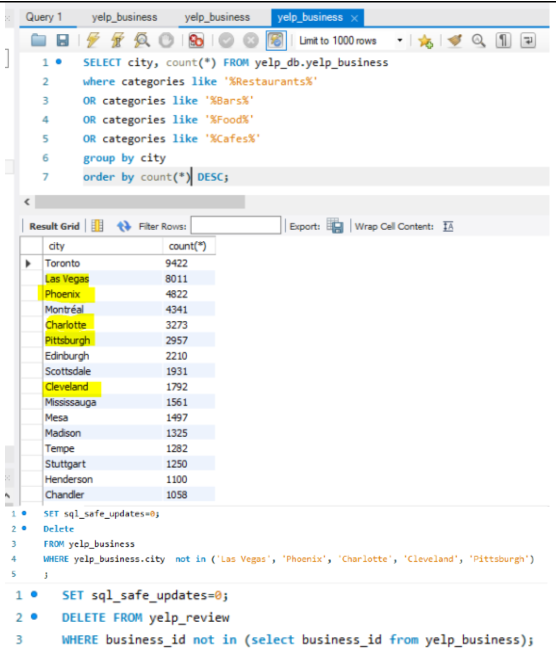


Figure 291: Yelp star rating of gastronomic businesses in Las Vegas

Appendix

I. Database Cleaning Operations



```
SELECT city, count(*) FROM yelp_db.yelp_business
where categories like '%Restaurants%'
OR categories like '%Bars%'
OR categories like '%Food%'
OR categories like '%Cafes%'
group by city
order by count(*) DESC;
```

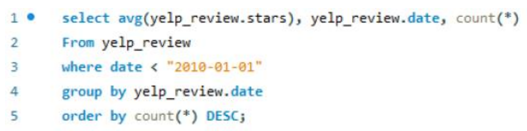
city	count(*)
Toronto	9422
Las Vegas	8011
Phoenix	4822
Montréal	4341
Charlotte	3273
Pittsburgh	2957
Edinburgh	2210
Scottsdale	1931
Cleveland	1792
Mississauga	1561
Mesa	1497
Madison	1325
Tempe	1282
Stuttgart	1250
Henderson	1100
Chandler	1058

```
SET sql_safe_updates=0;
Delete
FROM yelp_business
WHERE yelp_business.city  not in ('Las Vegas', 'Phoenix', 'Charlotte', 'Cleveland', 'Pittsburgh')
;
```

```
SET sql_safe_updates=0;
DELETE FROM yelp_review
WHERE business_id not in (select business_id from yelp_business);
```

In the Yelp data set, Toronto was the city with the most reviews about restaurants, café, bars, etc.
However, as we were interested in US cities, we select the first five US cities with the most reviews.

As can be seen, the database is extremely rich and has big potential. Unfortunately, we had to reduce the scope for project feasibility (time management), resource (not enough memory capacity on VM, SQL Server), and efficiency (SQL request can take up to 7 min to execute) .
In order to reduce our database size and be able to run queries in an efficient manner, we decided to create a copy table and delete all the reviews that are not about restaurants and food places, as well as only having information on the five US cities that we are interested in.



```
select avg(yelp_review.stars), yelp_review.date, count(*)
From yelp_review
where date < "2010-01-01"
group by yelp_review.date
order by count(*) DESC;
```

Deleting weather data for dates before 2008-01-01

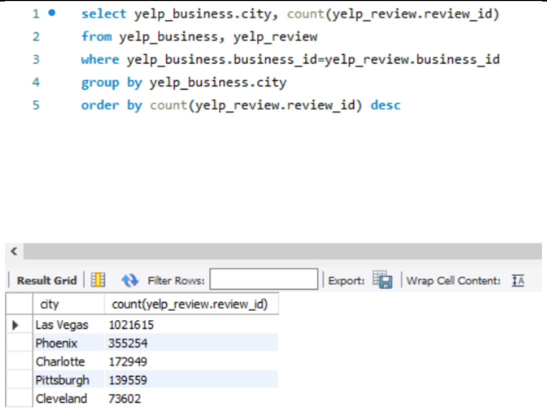
```
SET sql_safe_updates=0;

Delete

FROM weather_data

WHERE weather_data.date < "2008-01-01"
```

Deleting reviews for dates before 2008-01-01

```
SET sql_safe_updates=0;

Delete

FROM yelp_review

WHERE yelp_review.date < "2008-01-01"
```

Similarly, we investigated the date range, as initially we had reviews from 2003 till 2017. We checked the number of reviews for each year and found out that there were not many reviews before 2008 (<100 per year) while in 2014 and later there were thousands of reviews, which makes sense given the popularity of Yelp app came later in the 2010s. Therefore, we focused on the date range from 2008-2017 and deleted the information before 2008, similarly with weather data in order to reduce the size of our database furthermore.

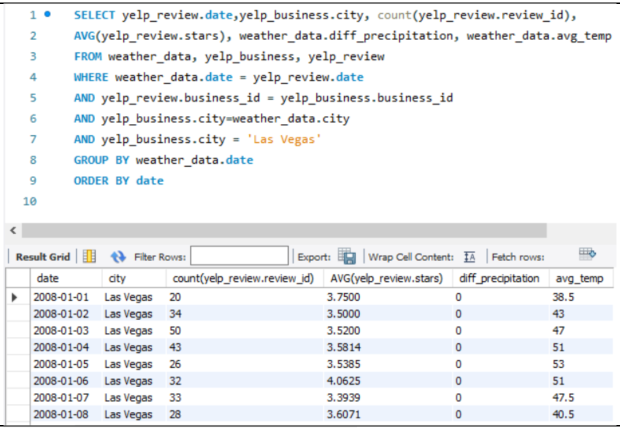
II. SQL request



```
select yelp_business.city, count(yelp_review.review_id)
from yelp_business, yelp_review
where yelp_business.business_id=yelp_review.business_id
group by yelp_business.city
order by count(yelp_review.review_id) desc
```

```
city        count(yelp_review.review_id)
Las Vegas   1021615
Phoenix     355254
Charlotte   172949
Pittsburgh  139559
Cleveland   73602
```

Potential problem of proportion in reviews.
Las Vegas has 10 times more reviews than other cities. Phoenix, Charlotte, Pittsburgh have a quite similar number of reviews.
Cleveland has the lowest amount of reviews.

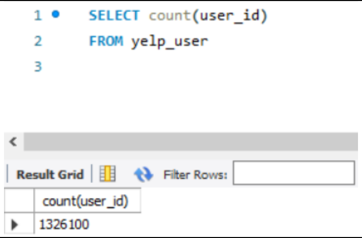


```sql
SELECT yelp_review.date,yelp_business.city, count(yelp_review.review_id),
AVG(yelp_review.stars), weather_data.diff_precipitation, weather_data.avg_temp
FROM weather_data, yelp_business, yelp_review
WHERE weather_data.date = yelp_review.date
AND yelp_review.business_id = yelp_business.business_id
AND yelp_business.city=weather_data.city
AND yelp_business.city = 'Las Vegas'
GROUP BY weather_data.date
ORDER BY date
```

date	city	count(yelp_review.review_id)	AVG(yelp_review.stars)	diff_precipitation	avg_temp
2008-01-01	Las Vegas	20	3.7500	0	38.5
2008-01-02	Las Vegas	34	3.5000	0	43
2008-01-03	Las Vegas	50	3.5200	0	47
2008-01-04	Las Vegas	43	3.5814	0	51
2008-01-05	Las Vegas	26	3.5385	0	53
2008-01-06	Las Vegas	32	4.0625	0	51
2008-01-07	Las Vegas	33	3.3939	0	47.5
2008-01-08	Las Vegas	28	3.6071	0	40.5

PyCharm and SQL connector: Get daily data about every city

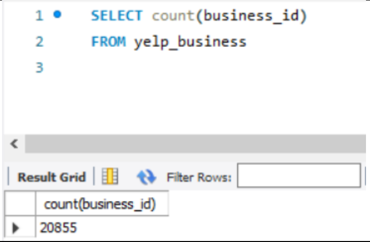
Connect reviews table with business table and weather data, count and average the reviews on daily basis.



```sql
SELECT count(user_id)
FROM yelp_user
```

count(user_id)
1326100

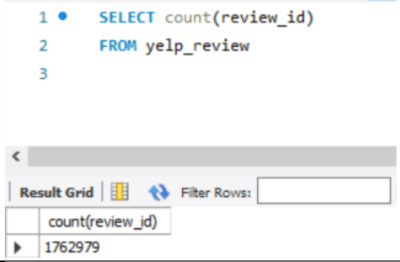
1.3 million users



```sql
SELECT count(business_id)
FROM yelp_business
```

count(business_id)
20855

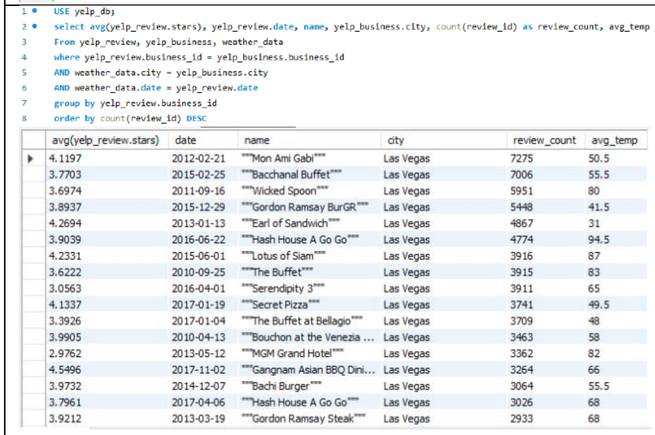
21’000 businesses



```sql
SELECT count(review_id)
FROM yelp_review
```

count(review_id)
1762979

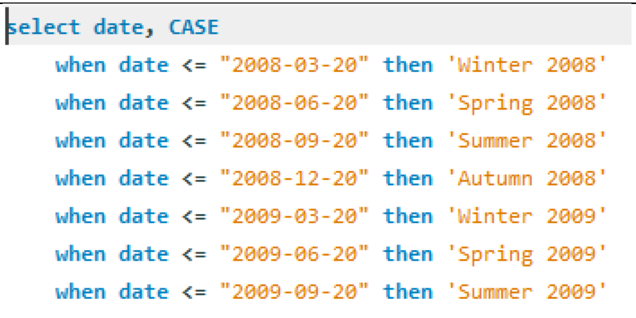
1.8 million reviews



```sql
USE yelp_db;
select avg(yelp_review.stars), yelp_review.date, name, yelp_business.city, count(review_id) as review_count, avg_temp
From yelp_review, yelp_business, weather_data
where yelp_review.business_id = yelp_business.business_id
AND weather_data.city = yelp_business.city
AND weather_data.date = yelp_review.date
group by yelp_review.business_id
order by count(review_id) DESC
```

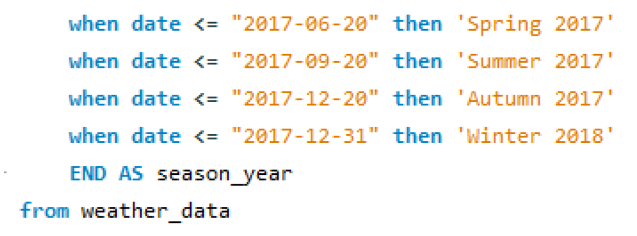
avg(yelp_review.stars)	date	name	city	review_count	avg_temp
4.1197	2012-02-21	"""Mon Ami Gabi"""	Las Vegas	7275	50.5
3.7703	2015-02-25	"""Bacchanal Buffet"""	Las Vegas	7006	55.5
3.6974	2011-09-16	"""Wicked Spoon"""	Las Vegas	5951	80
3.8937	2015-12-29	"""Gordon Ramsay BurGR"""	Las Vegas	5448	41.5
4.2694	2013-01-13	"""Earl of Sandwich"""	Las Vegas	4867	31
3.9039	2016-06-22	"""Hash House A Go Go"""	Las Vegas	4774	94.5
4.2331	2015-06-01	"""Lotus of Siam"""	Las Vegas	3916	87
3.6222	2010-09-25	"""The Buffet"""	Las Vegas	3915	83
3.0563	2016-04-01	"""Serendipity 3"""	Las Vegas	3911	65
4.1337	2017-01-19	"""Secret Pizza"""	Las Vegas	3741	49.5
3.3926	2017-01-04	"""The Buffet at Bellagio"""	Las Vegas	3709	48
3.9905	2010-04-13	"""Bouchon at the Venezia ...	Las Vegas	3463	58
2.9762	2013-05-12	"""MGM Grand Hotel"""	Las Vegas	3362	82
4.5496	2017-11-02	"""Gangnam Asian BBQ Dini...	Las Vegas	3264	66
3.9732	2014-12-07	"""Bachi Burger"""	Las Vegas	3064	55.5
3.7961	2017-04-06	"""Hash House A Go Go"""	Las Vegas	3026	68
3.9212	2013-03-19	"""Gordon Ramsay Steak"""	Las Vegas	2933	68

Finding out the average rating and number of reviews for each restaurant on a given date and connecting the weather data so we get the temperature and precipitation on that given date to analyse whether weather has an impact on the user rating.

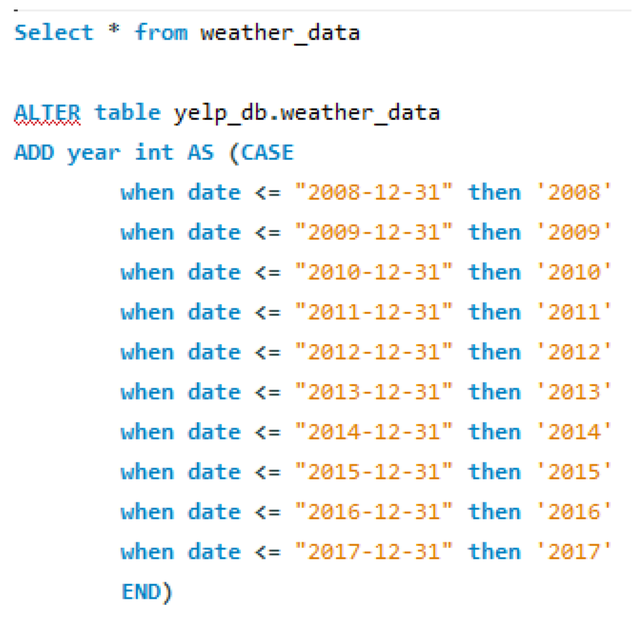


```
select date, CASE
    when date <= "2008-03-20" then 'Winter 2008'
    when date <= "2008-06-20" then 'Spring 2008'
    when date <= "2008-09-20" then 'Summer 2008'
    when date <= "2008-12-20" then 'Autumn 2008'
    when date <= "2009-03-20" then 'Winter 2009'
    when date <= "2009-06-20" then 'Spring 2009'
    when date <= "2009-09-20" then 'Summer 2009'
```

….. and so till 2017….



```
    when date <= "2017-06-20" then 'Spring 2017'
    when date <= "2017-09-20" then 'Summer 2017'
    when date <= "2017-12-20" then 'Autumn 2017'
    when date <= "2017-12-31" then 'Winter 2018'
    END AS season_year
from weather_data
```



```
Select * from weather_data

ALTER table yelp_db.weather_data
ADD year int AS (CASE
        when date <= "2008-12-31" then '2008'
        when date <= "2009-12-31" then '2009'
        when date <= "2010-12-31" then '2010'
        when date <= "2011-12-31" then '2011'
        when date <= "2012-12-31" then '2012'
        when date <= "2013-12-31" then '2013'
        when date <= "2014-12-31" then '2014'
        when date <= "2015-12-31" then '2015'
        when date <= "2016-12-31" then '2016'
        when date <= "2017-12-31" then '2017'
        END)
```

Weather data table is altered to add columns for year and seasons in order to investigate whether rating is affected by seasonality and whether restaurants rating improved or decreased over the years. That was achieved with command "CASE"

--Average rating per season

select yelp_business.city, avg(yelp_review.stars), season

from yelp_business, weather_data, yelp_review

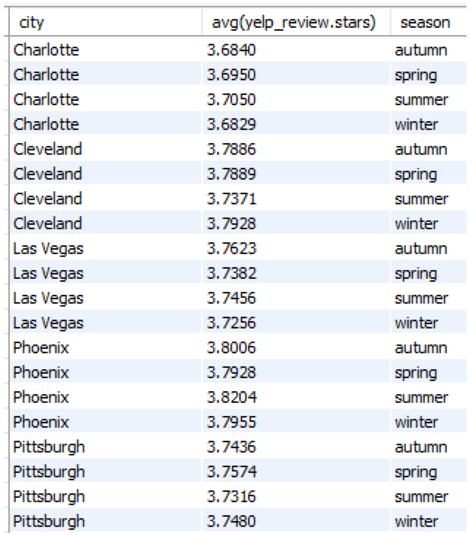
where yelp_business.city = weather_data.city

and yelp_review.business_id = yelp_business.business_id

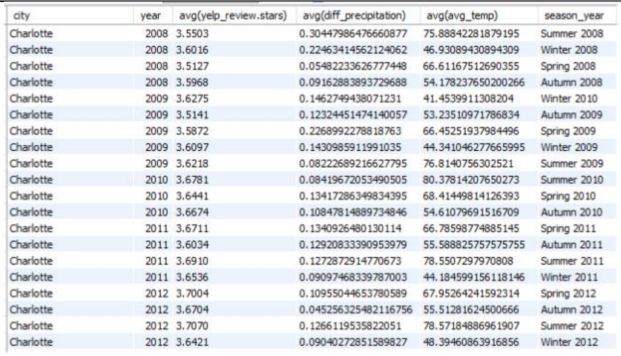
and weather_data.date = yelp_review.date

Group by city, season

Order by city



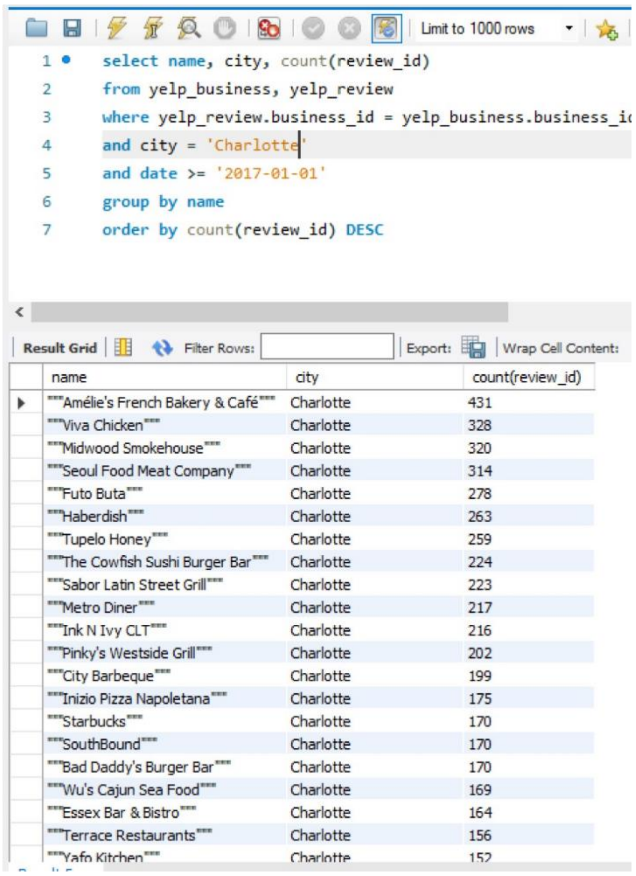
city	avg(yelp_review.stars)	season
Charlotte	3.6840	autumn
Charlotte	3.6950	spring
Charlotte	3.7050	summer
Charlotte	3.6829	winter
Cleveland	3.7886	autumn
Cleveland	3.7889	spring
Cleveland	3.7371	summer
Cleveland	3.7928	winter
Las Vegas	3.7623	autumn
Las Vegas	3.7382	spring
Las Vegas	3.7456	summer
Las Vegas	3.7256	winter
Phoenix	3.8006	autumn
Phoenix	3.7928	spring
Phoenix	3.8204	summer
Phoenix	3.7955	winter
Pittsburgh	3.7436	autumn
Pittsburgh	3.7574	spring
Pittsburgh	3.7316	summer
Pittsburgh	3.7480	winter



city	year	avg(yelp_review.stars)	avg(diff_precipitation)	avg(avg_temp)	season_year
Charlotte	2008	3.5503	0.30447986476660877	75.88842281879195	Summer 2008
Charlotte	2008	3.6016	0.22463414562124062	46.93089430894309	Winter 2008
Charlotte	2008	3.5127	0.05482233626777448	66.61167512690355	Spring 2008
Charlotte	2008	3.5968	0.09162883893729688	54.178237650200266	Autumn 2008
Charlotte	2009	3.6275	0.1462749438071231	41.4539911308204	Winter 2010
Charlotte	2009	3.5141	0.12324451474140057	53.23510971786834	Autumn 2009
Charlotte	2009	3.5872	0.2268992278818763	66.45251937984496	Spring 2009
Charlotte	2009	3.6097	0.1430985911991035	44.341046277665995	Winter 2009
Charlotte	2009	3.6218	0.08222689216627795	76.8140756302521	Summer 2009
Charlotte	2010	3.6781	0.08419672053490505	80.37814207650273	Summer 2010
Charlotte	2010	3.6441	0.13417286349834395	68.41449814126393	Spring 2010
Charlotte	2010	3.6674	0.10847814889734846	54.61079691516709	Autumn 2010
Charlotte	2011	3.6711	0.1340926480130114	66.78598774885145	Spring 2011
Charlotte	2011	3.6034	0.12920833390953979	55.588825757575755	Autumn 2011
Charlotte	2011	3.6910	0.1272872914770673	78.5507297970808	Summer 2011
Charlotte	2011	3.6536	0.09097468339787003	44.184599156118146	Winter 2011
Charlotte	2012	3.7004	0.10955044653780589	67.95264241592314	Spring 2012
Charlotte	2012	3.6704	0.045256325482116756	55.51281624500666	Autumn 2012
Charlotte	2012	3.7070	0.1266119535822051	78.57184886961907	Summer 2012
Charlotte	2012	3.6421	0.09040272851589827	48.39460863916856	Winter 2012

Investigating the rating in each season of each year, connected with average temperature and precipitation corresponding to that season of the year.

Looking into the most reviewed restaurants in each city.



```
select name, city, count(review_id)
from yelp_business, yelp_review
where yelp_review.business_id = yelp_business.business_id
and city = 'Charlotte'
and date >= '2017-01-01'
group by name
order by count(review_id) DESC
```

name	city	count(review_id)
"""Amélie's French Bakery & Café"""	Charlotte	431
"""Viva Chicken"""	Charlotte	328
"""Midwood Smokehouse"""	Charlotte	320
"""Seoul Food Meat Company"""	Charlotte	314
"""Futo Buta"""	Charlotte	278
"""Haberdish"""	Charlotte	263
"""Tupelo Honey"""	Charlotte	259
"""The Cowfish Sushi Burger Bar"""	Charlotte	224
"""Sabor Latin Street Grill"""	Charlotte	223
"""Metro Diner"""	Charlotte	217
"""Ink N Ivy CLT"""	Charlotte	216
"""Pinky's Westside Grill"""	Charlotte	202
"""City Barbeque"""	Charlotte	199
"""Inizio Pizza Napoletana"""	Charlotte	175
"""Starbucks"""	Charlotte	170
"""SouthBound"""	Charlotte	170
"""Bad Daddy's Burger Bar"""	Charlotte	170
"""Wu's Cajun Sea Food"""	Charlotte	169
"""Essex Bar & Bistro"""	Charlotte	164
"""Terrace Restaurants"""	Charlotte	156
"""Yafo Kitchen"""	Charlotte	152

Getting the restaurants all cities have in common to analyse whether the rating differs for the same restaurant from one city to another. It is a bit tricky with mySQL as it does not use 'INTERSECT' but rather IN (select...)

select name, count(review_id)

from yelp_business, yelp_review

where city = 'Las Vegas'

And name IN (select name from yelp_business

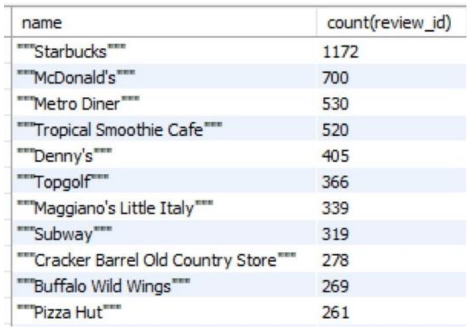
where city = 'Charlotte')

and yelp_review.business_id = yelp_business.business_id

and date >= '2017-01-01'

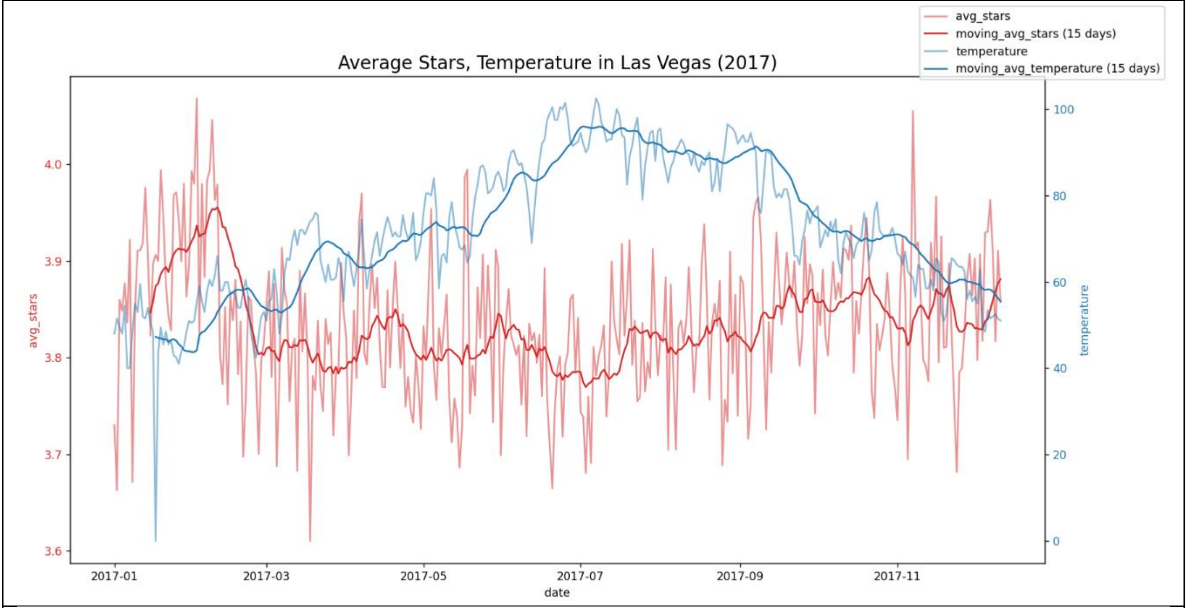
group by name

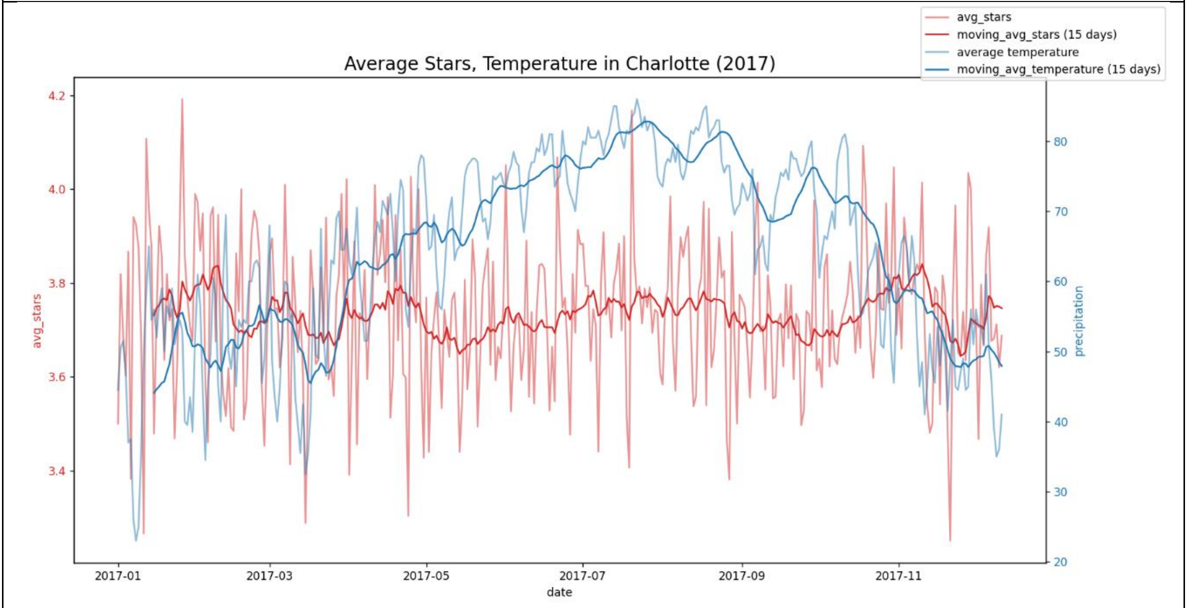
order by count(review_id) DESC



name	count(review_id)
"""Starbucks"""	1172
"""McDonald's"""	700
"""Metro Diner"""	530
"""Tropical Smoothie Cafe"""	520
"""Denny's"""	405
"""Topgolf"""	366
"""Maggiano's Little Italy"""	339
"""Subway"""	319
"""Cracker Barrel Old Country Store"""	278
"""Buffalo Wild Wings"""	269
"""Pizza Hut"""	261

III. Complement to Analysis





Temperature for Las Vegas and Charlotte for the year 2017