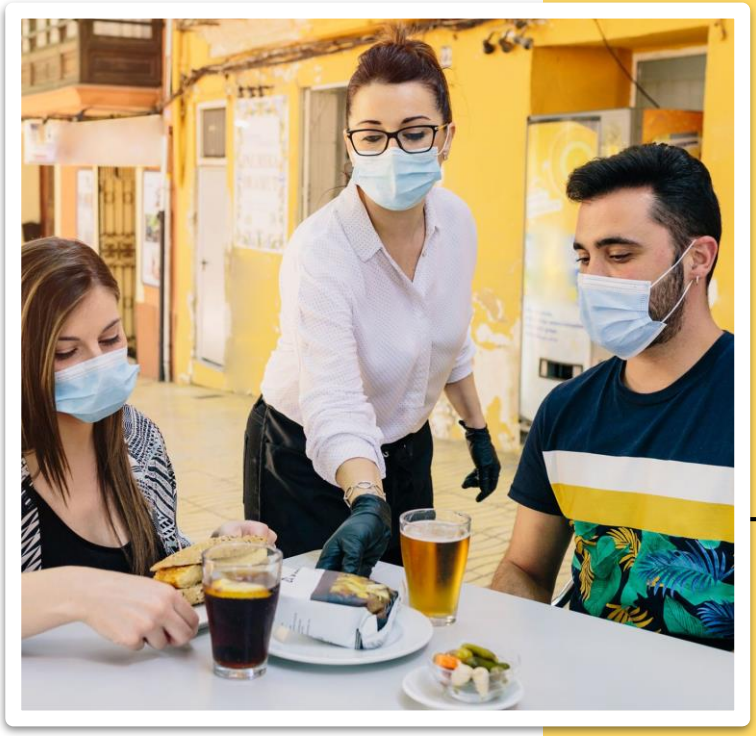


COVID-19 Restaurant Analysis

December 3, 2020

TABLE OF CONTENTS

01 Executive Summary

The entire restaurant industry has changed and it becomes harder for the restaurants to survive due to Covid-19. They have to make some changes such as limiting the dine-in capacity and asking the staff to wear the masks all the time. People are facing the problem about finding the restaurants that not only have the best food, but also provide the safest environment. **Therefore, our project aims to help the customers to find the best and safest restaurants in the Las Vegas area.**

We plan to analyze each restaurant in Las Vegas through collecting the safety related data and the review data for the restaurants from Yelp and Zomato using external APIs then processing those datasets using advanced python packages. We also plan to use Tableau for data visualization purposes.

02 Business Use Case

Situation

COVID-19 has changed the entire bar/restaurant industry. Extra rules and guidelines have been put in place for restaurants in Las Vegas, adding even more factors into customers' decisions when considering where to dine. Many people want to support their favorite restaurants and local businesses, while still remaining safe and making informed choices.

Question

Can we **rate restaurants** based on their health/safety features and customer ratings since COVID started in order to **find the safest and best restaurants to dine at in Las Vegas**?

Approach

- Collect relevant safety data and reviews from **Yelp & Zomato** then organize it into database
- **Analyze** each restaurant based on customer reviews as well and incorporates the health/safety offerings of the restaurant

03

DATA PROFILE

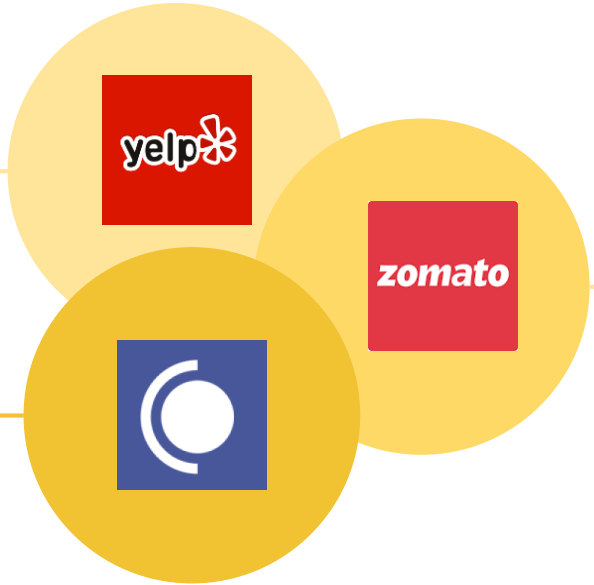
03 Data Sources

Yelp Open Dataset

contains 8 million reviews and 210,000 businesses around 10 metropolitan areas.

COVID Tracking Project API

Provides access to daily national and state level COVID-19 related metrics.



Zomato API

returns rating, location, & review data for up to 100 restaurants

03 Data Profile

	Yelp Business	Yelp COVID	Zomato	COVID Tracking Project
Size	1.5MB		11KB	23KB
Quantities	4,386 observations 18 attributes	4,386 observations 9 attributes	100 observations 6 attributes	253 observations 13 attributes
Format	JSON structured	JSON structured	JSON structured	JSON structured
Matching Criteria	Join with Zomato using xref_key - restaurant name + address	Join with Yelp Business on business_id	Join with Yelp using xref_key - restaurant name + address	Join with restaurant data by state

Data profiles are based on datasets after cleaning & transforming for use

03 Data Quality

COMPLETENESS

Are values missing?

The yelp dataset had issues with names missing or no data in the delivery/takeout flags. This data could be manually updated with more time.

VALIDITY

Does data match the rules?

All fields were checked and formatted to be the appropriate data type in our database.

UNIQUENESS

Is there duplicated data?

All restaurants are unique. Categories are duplicated across restaurants but de-duped in the categories table.

CONSISTENCY

Consistent across various data stores?

All data is stored in and sourced from GCP, so it is consistent for all users in MySQL & Tableau.

TIMELINESS

Does data represent reality from required point in time?

Yelp's dataset is old, ideally we would like to access the API to pull recent review data from COVID time period, but we were limited to their academic dataset.
Zomato & COVID data is recent from their APIs.

ACCURACY

Degree to which data represents reality

Restaurant ratings are only based on people who submit reviews and can be biased if they had a particularly positive or negative experience.

COVID data is only based on those who get tested and may not be reflective of actual cases.

METHODOLOGY & TOOLS

04

04 Tools

Extract, Transform, Load

Python via **Jupyter Notebook** were used for the analysis and transformation of our data into a usable form. Numpy and pandas packages were used to process, organize and clean the data.

OpenRefine was used to clean the data and libraries.

Storage

Google Cloud Platform and **MySQL Workbench** were used to create a relational database and to store our data.

Insights & Visualization

Visualizations of the data are shown with **Tableau**.

04 Data Preparation

EXTRACT

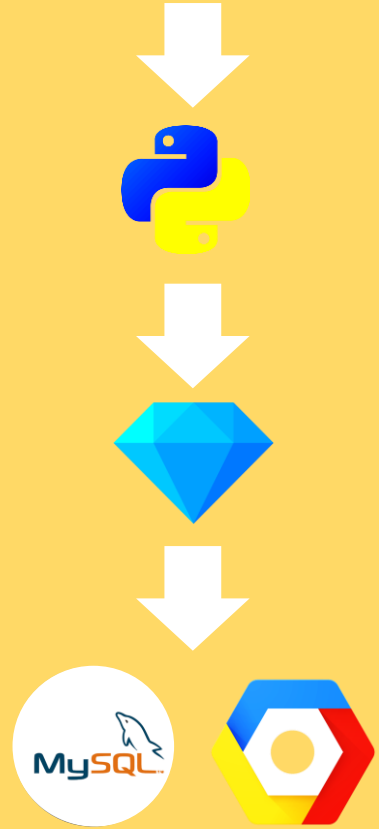
- ➔ Extract restaurant in Las Vegas data from Zomato using its API
- ➔ Extract Nevada Covid-19 daily case data from covidtracking.com using its API
- ➔ Download restaurant & COVID dataset from Yelp

TRANSFORM

- ➔ Use python to create organized dataframes and select relevant attributes & rows
- ➔ Use OpenRefine to clean the datasets

LOAD

- ➔ Use MySQL to connect tables and build Enhanced Entity Relationship model
- ➔ Import data into Google Cloud Platform



Database Design Considerations

Our **OLAP dimensional** database will utilize a **snowflake schema** for ease of analysis & reporting

01

Subject-Oriented

This database will be used for the purpose of helping customers find the right restaurant to dine at while COVID precautions are in place

02

Multi-Dimensional

Dimensions were determined by attributes with duplicates across restaurants, such as categories, then placed in a separate table and given an ID

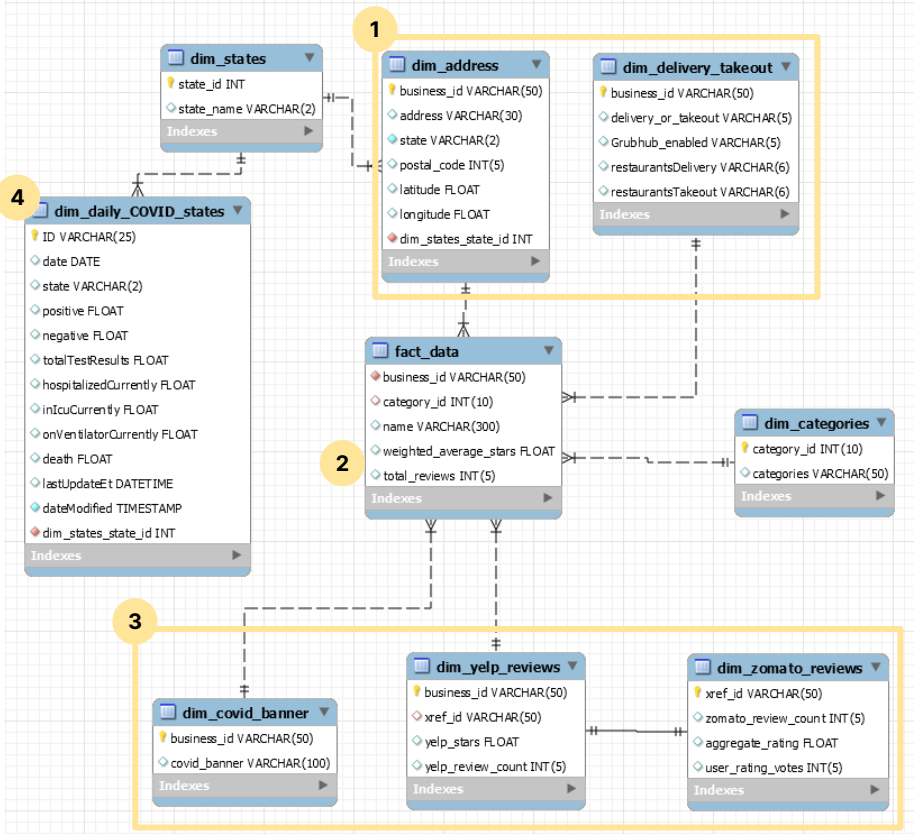
A snowflake schema was selected due one dimension being connected to another

03

High Granularity

The fact table contains multiple rows for each restaurant and each of its features in the dimension tables

Enhanced Entity Relationship Model



DESIGN CONSIDERATIONS

1. **Address & delivery/takeout attributes** are likely to remain **static** for long periods of time, so they sit in their own tables

1. **Weighted average stars and total reviews** are **calculated measures** based on data from both Yelp & Zomato

1. **COVID & Yelp/Zomato review** data is **dynamic**, so we stored them in separate tables to make frequent updates easier

1. While our database only includes data for Las Vegas, we built the COVID and states tables to accommodate more states with the **assumption this database would extend to more states when rolled out nationwide**

04 NoSQL Database

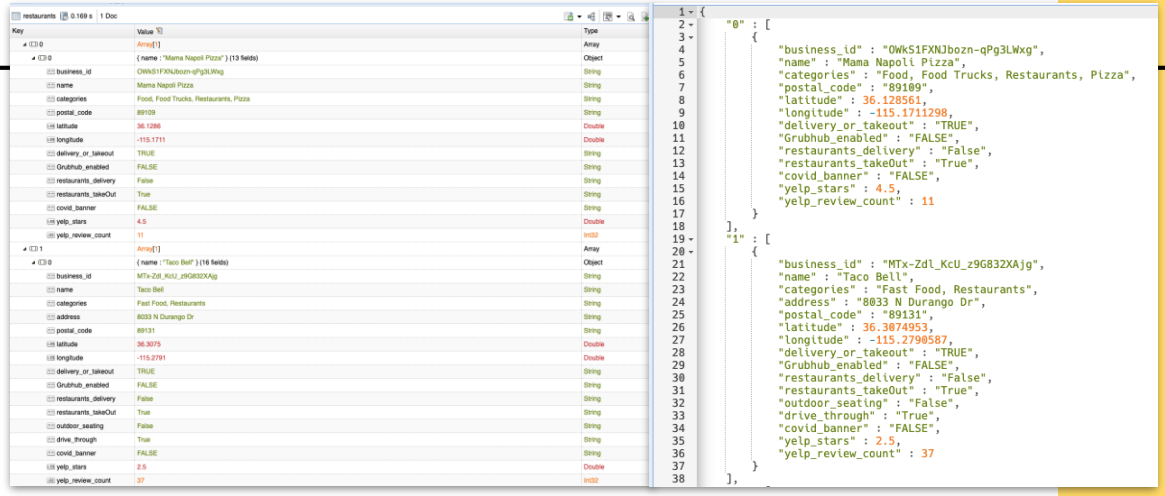
For document-oriented NoSQL databases like MongoDB, **each "row" of a restaurant shall be a "document"**

- In our relational database the structure is fixed and each restaurant row needs to adhere to the table structure, but in our document-based databases the **structure is loosely coupled**

Since our use case is to identify restaurants with their respective rating and COVID measures, **we consider two tables are enough**:

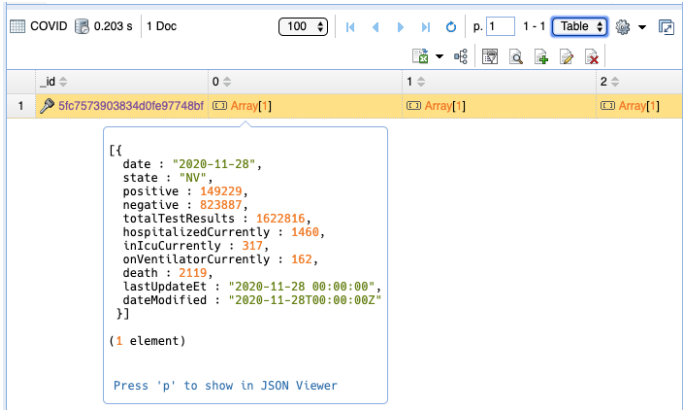
- Table 1: **Restaurant** dataset (biz_id, name, reviews, categories, address, etc...)
- Table 2: **Nevada COVID** dataset

Restaurant dataset in Tree & Json format




```
{
  "0" : [
    {
      "business_id" : "OWkS1FXNJbozn-qPg3LWxg",
      "name" : "Mama Napoli Pizza",
      "categories" : "Food, Food Trucks, Restaurants, Pizza",
      "postal_code" : "89109",
      "latitude" : 36.128561,
      "longitude" : -115.1711298,
      "delivery_or_takeout" : "TRUE",
      "Grubhub_enabled" : "FALSE",
      "restaurants_delivery" : "False",
      "restaurants_takeOut" : "True",
      "covid_banner" : "FALSE",
      "yelp_stars" : 4.5,
      "yelp_review_count" : 11
    }
  ],
  "1" : [
    {
      "business_id" : "MTx-Zdl_KcU_z9G832XAjg",
      "name" : "Taco Bell",
      "categories" : "Fast Food, Restaurants",
      "address" : "8033 N Durango Dr",
      "postal_code" : "89131",
      "latitude" : 36.3074953,
      "longitude" : -115.2790587,
      "delivery_or_takeout" : "TRUE",
      "Grubhub_enabled" : "FALSE",
      "restaurants_delivery" : "False",
      "restaurants_takeOut" : "True",
      "outdoor_seating" : "False",
      "drive_through" : "True",
      "covid_banner" : "FALSE",
      "yelp_stars" : 2.5,
      "yelp_review_count" : 37
    }
  ],
```


Nevada COVID dataset in Table format

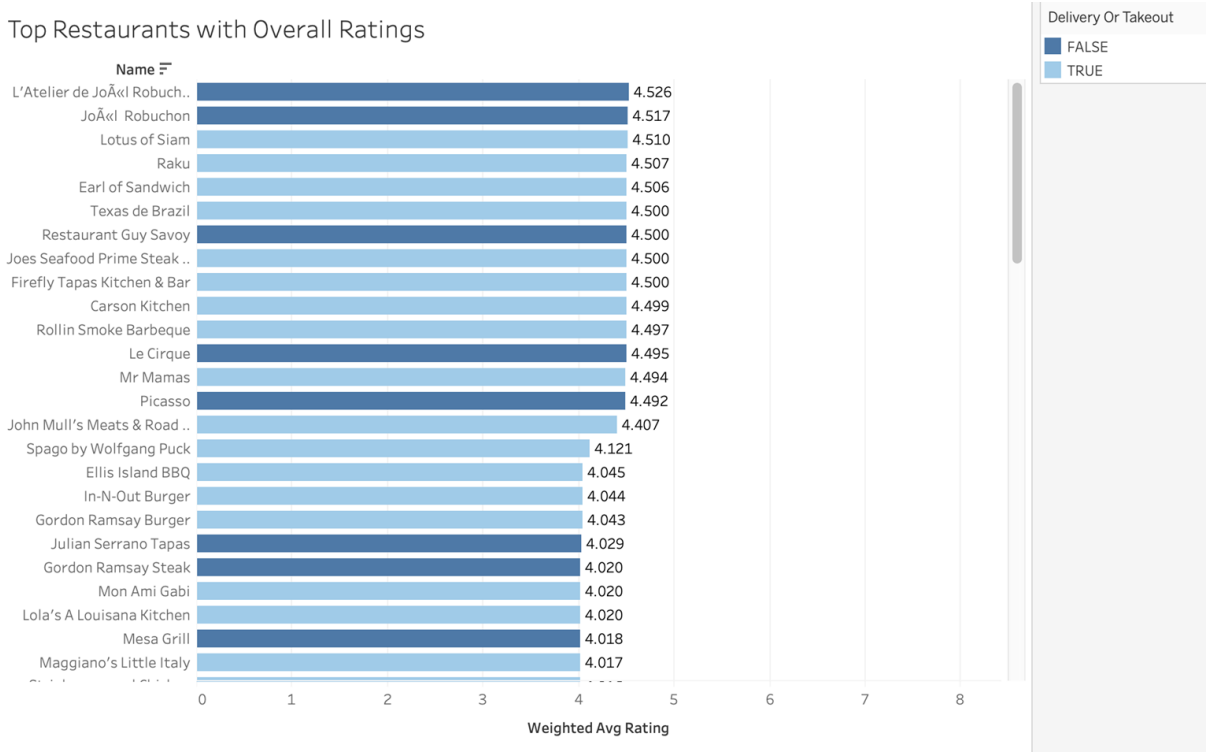



```
[{
  date : "2020-11-28",
  state : "NV",
  positive : 149229,
  negative : 823887,
  totalTestResults : 1622816,
  hospitalizedCurrently : 1460,
  inIcuCurrently : 317,
  onVentilatorCurrently : 162,
  death : 2119,
  lastUpdateEt : "2020-11-28 00:00:00",
  dateModified : "2020-11-28T00:00:00Z"
}]

(1 element)

Press 'p' to show in JSON Viewer
```

05

INSIGHTS

Top Las Vegas Restaurants (Overall Ratings)



INSIGHT

The top three restaurants that have the highest weighted average score from the Yelp score and the Zomato score are:

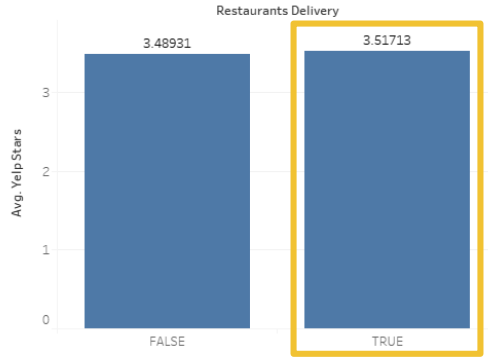
1. L'Atelier de Joel Robuchon
2. Joel Robuchon
3. Lotus of Siam.

Most of the top-rated restaurants provide such services.

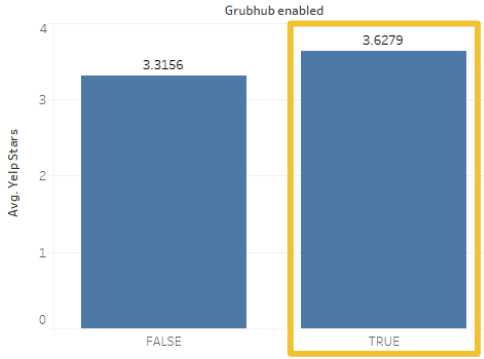
Attribute Effects on Ratings (Yelp)

Yelp Rating by Attributes

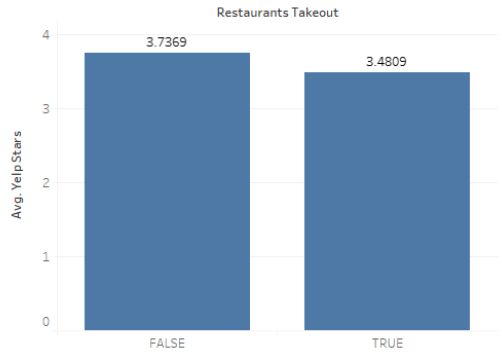
Delivery Rating



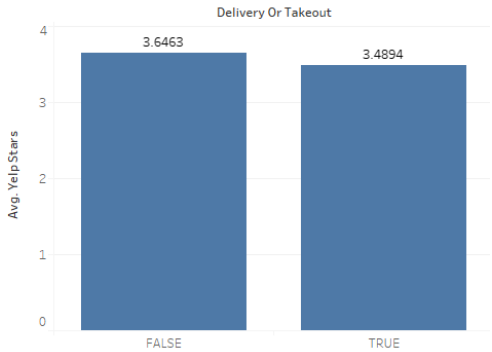
Grubhub



Takeout Rating

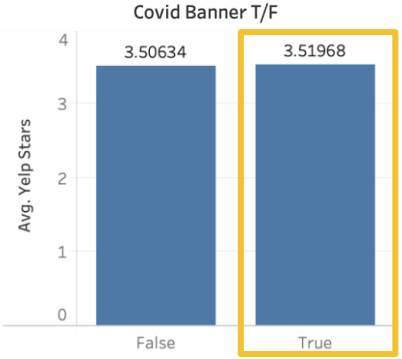


Delivery_Takeout Rating

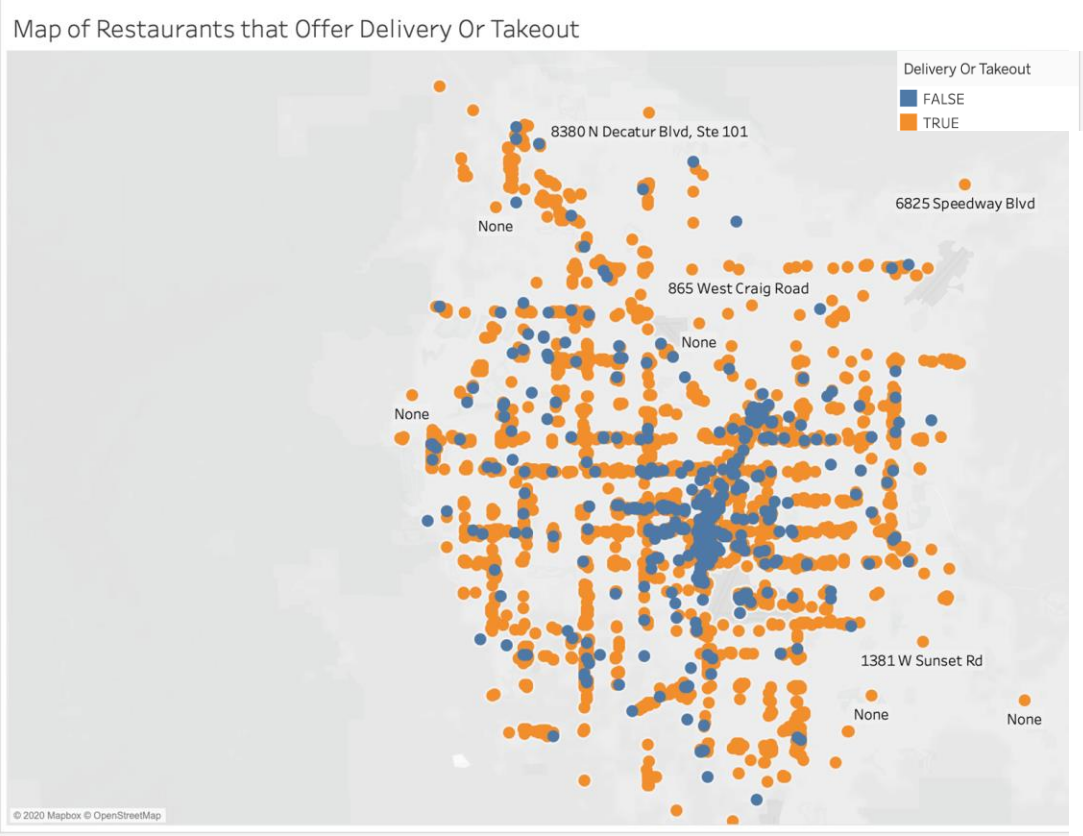


INSIGHT

Restaurants with delivery & grubhub enabled and a COVID banner have higher ratings on Yelp



05 Map of Restaurants that Offer Delivery Or Takeout

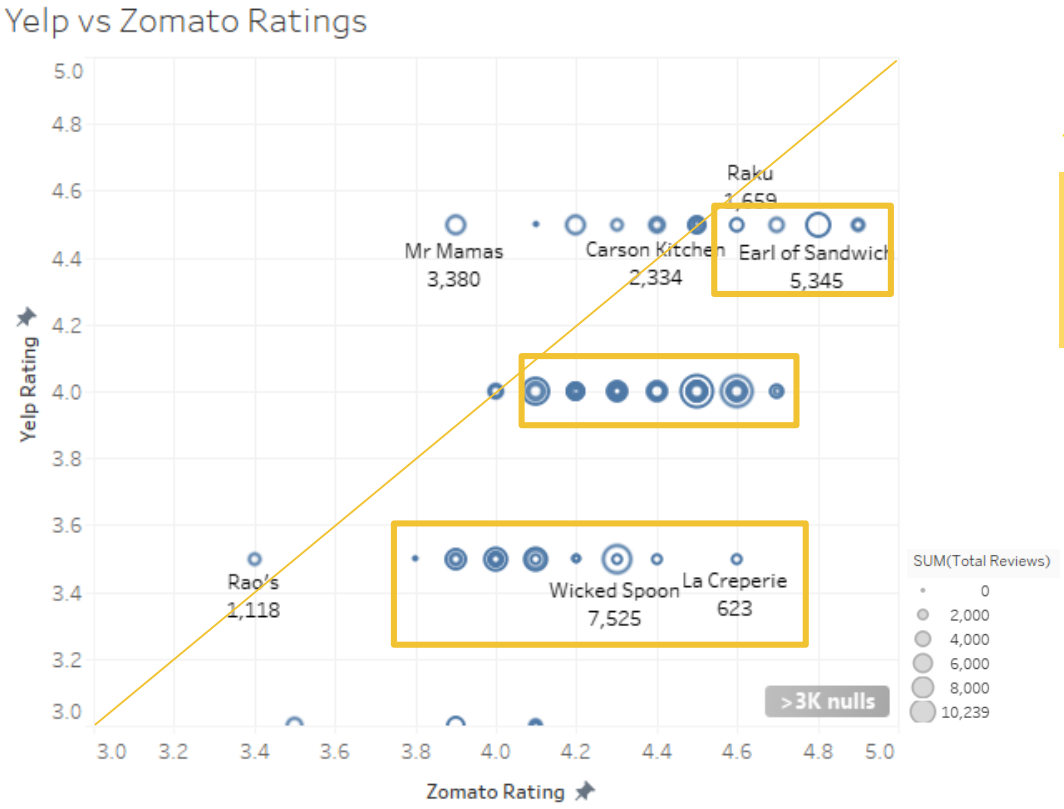


INSIGHT

Many restaurants in downtown Las Vegas (center of the map) do not offer delivery or takeout.

This is expected because many of these restaurants are in the hotels on the strip, and only cater to hotel guests.

05 Comparing Yelp & Zomato Ratings



INSIGHT

Zomato Ratings tend to skew higher than Yelp Ratings

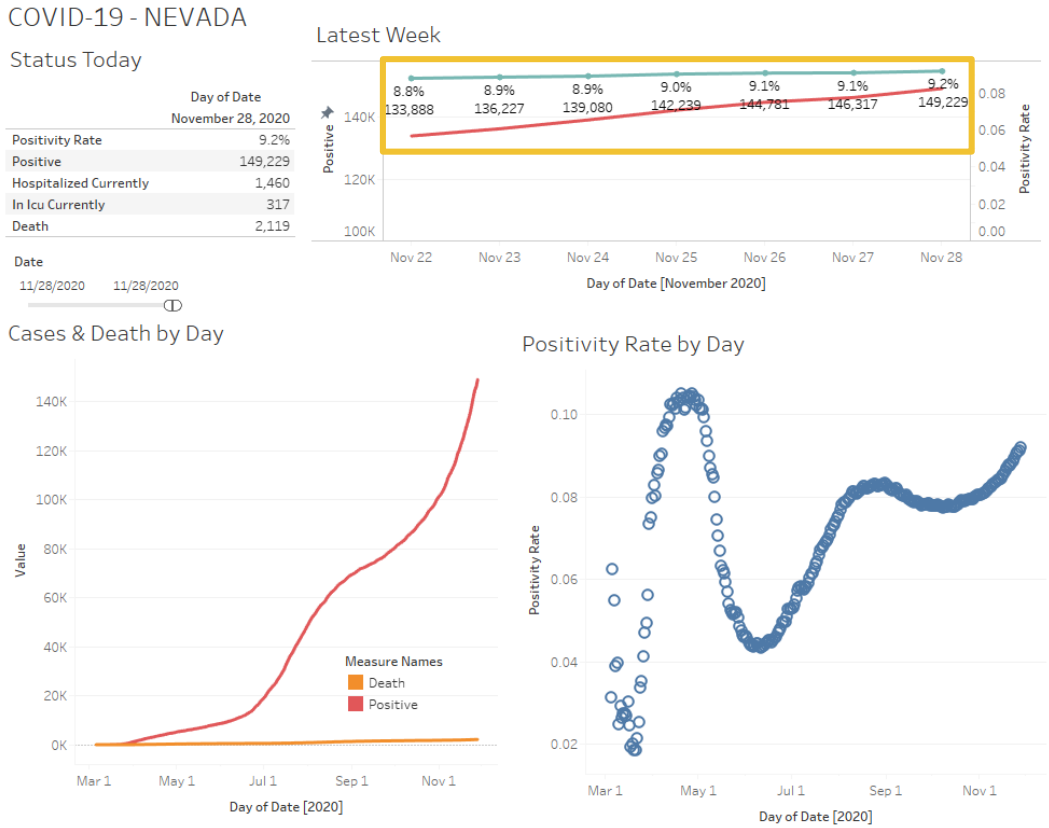
COVID-19 Dashboard - Nevada

In addition to restaurant data, we created a dashboard summarizing the **state of the pandemic in Nevada** for customers to keep up to date on the **latest trends**.

By including this data, we hope customers will make a more **informed decision** about how they will dine with a restaurant (inside, outside, delivery, etc.).

INSIGHT

Positivity rate has been on the rise the over the last 7 days, customers should take precaution when dining out



RECOMMENDATIONS & LESSONS LEARNED

06

Corrective Measures

Cleaned, transformed, & formatted data:

Issues	Solutions
Yelp Dataset: Multiple categories were listed in one column for each restaurant and duplicated across rows	● Created separate table with distinct category entries and category id ● Parsed out categories for each restaurant and created new row for each restaurant/category combination in fact table
Table with restaurant names was not loading completely	Some restaurant names had commas in them and needed to be removed
Missing values in our dataset due to the difficulty of data collection	In NoSQL database, we eliminated the values that are NaN or empty in the dataframe and only include valuable information in the JSON data models

Scope for Improvements

Create restaurant safety scores

Calculate score for each restaurant incorporating ratings, reviews, and COVID features

Collect reviews from March 2020 forward

Analyze reviews for sentiment towards restaurants' safety precautions

Obtain COVID case data by zip code

Observe if increases in cases affect restaurant ratings in certain areas

Collect ratings weekly & store historical data

Analyze how ratings trend over time and if safety precautions and COVID case trends impact ratings

06 Lessons Learned

WHAT DID WE LEARN?

- **Data Cleaning is required**
  - Even large established companies like Yelp have messy data and require cleaning before using in analysis
- **Data Extraction Limitations**
  - Many review websites have limitations around collecting large amounts of data from their APIs, which made finding data for this use case difficult
- **Referential Integrity**
  - Data will not load into a child table with a foreign key constraint if the parent table does not have a matching value
  - Issues with the load of the parent table data was discovered when the fact_data table (child) would not load

07 OUR TEAM

Yue Shen

Anna Willman

Collin Li

Angela Zheng

Thank you!

Questions?

08

APPENDIX

References

Yelp: https://www.yelp.com/dataset/download

Zomato: https://developers.zomato.com/api

COVID Tracking Project: https://covidtracking.com/data/api

Data Profile - Yelp (Business & COVID)


```
df_biz[df_biz.city == 'Las Vegas'].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 31631 entries, 6 to 209386
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   business_id    31631 non-null  object 
 1   name           31631 non-null  object 
 2   address        31631 non-null  object 
 3   city           31631 non-null  object 
 4   state          31631 non-null  object 
 5   postal_code    31631 non-null  object 
 6   latitude       31631 non-null  float64
 7   longitude      31631 non-null  float64
 8   stars          31631 non-null  float64
 9   review_count   31631 non-null  int64  
 10  is_open        31631 non-null  int64  
 11  attributes     27500 non-null  object 
 12  categories     31556 non-null  object 
 13  hours          25041 non-null  object 
dtypes: float64(3), int64(2), object(9)
memory usage: 3.6+ MB
```

```
df_covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209795 entries, 0 to 209794
Data columns (total 9 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   business_id                 209795 non-null  object 
 1   highlights                  209795 non-null  object 
 2   delivery or takeout         209795 non-null  object 
 3   Grubhub enabled             209795 non-null  object 
 4   Call To Action enabled      209795 non-null  object 
 5   Request a Quote Enabled     209795 non-null  object 
 6   Covid Banner                209795 non-null  object 
 7   Temporary Closed Until      209795 non-null  object 
 8   Virtual Services Offered    209795 non-null  object 
```

Data Profile - FourSquare (100 requests)


```
df_foursquare = pd.json_normalize(data['response']['groups'][0]['items'])
df_foursquare.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 39 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   referralId                            100 non-null    object
 1   reasons.count                         100 non-null    int64
 2   reasons.items                         100 non-null    object
 3   venue.id                              100 non-null    object
 4   venue.name                            100 non-null    object
 5   venue.location.address                100 non-null    object
 6   venue.location.crossStreet            71 non-null     object
 7   venue.location.lat                    100 non-null    float64
 8   venue.location.lng                    100 non-null    float64
 9   venue.location.labeledLatLngs         100 non-null    object
 10  venue.location.postalCode             99 non-null     object
 11  venue.location.cc                     100 non-null    object
 12  venue.location.city                   100 non-null    object
 13  venue.location.state                  100 non-null    object
 14  venue.location.country                100 non-null    object
 15  venue.location.formattedAddress       100 non-null    object
 16  venue.categories                      100 non-null    object
 17  venue.verified                        100 non-null    bool
 18  venue.stats.tipCount                  100 non-null    int64
 19  venue.stats.usersCount                100 non-null    int64
 20  venue.stats.checkinsCount             100 non-null    int64
 21  venue.stats.visitsCount               100 non-null    int64
 22  venue.beenHere.count                  100 non-null    int64
 23  venue.beenHere.lastCheckinExpiredAt   100 non-null    int64
 24  venue.beenHere.marked                 100 non-null    bool
 25  venue.beenHere.unconfirmedCount       100 non-null    int64
 26  venue.photos.count                    100 non-null    int64
 27  venue.photos.groups                   100 non-null    object
 28  venue.hereNow.count                   100 non-null    int64
 29  venue.hereNow.summary                 100 non-null    object
 30  venue.hereNow.groups                  100 non-null    object
 31  venue.delivery.id                     45 non-null     object
 32  venue.delivery.url                    45 non-null     object
 33  venue.delivery.provider.name          45 non-null     object
 34  venue.delivery.provider.icon.prefix   45 non-null     object
 35  venue.delivery.provider.icon.sizes    45 non-null     object
 36  venue.delivery.provider.icon.name     45 non-null     object
 37  venue.venuePage.id                    25 non-null     object
 38  venue.location.neighborhood           9 non-null      object
dtypes: bool(2), float64(2), int64(10), object(25)
memory usage: 29.2+ KB
```

ETL- download relevant datasets from Yelp

YELP DATASETS

```
df_biz = pd.read_json("yelp_academic_dataset_business.json", lines = True)
df_covid = pd.read_json("yelp_academic_dataset_covid_features.json", lines = True)
```

```
df_biz.head()
```

	address	attributes	business_id	categories	city	hours	is_open	latitude	longitude	name	postal_co
0	10913 Bailey Rd	{'BusinessAcceptsCreditCards': 'True', 'BikePa...	f9NumwFMBDn751xgFiRbNA	Active Life, Gun/Rifle Ranges, Guns & Ammo, Sh...	Cornelius	{'Monday': '10:0-18:0', 'Tuesday': '11:0-20:0'...	1	35.462724	-80.852612	The Range At Lake Norman	280
1	8880 E Via Linda, Ste 107	{'GoodForKids': 'True', 'ByAppointmentOnly': '...	Yzvjg0SayhoZgCljUJRF9Q	Health & Medical, Fitness & Instruction, Yoga,...	Scottsdale	None	1	33.569404	-111.890264	Carlos Santo, NMD	852
2	3554 Rue Notre-Dame O	None	XNoUzKckATkOD1hP6vghZg	Pets, Pet Services, Pet Groomers	Montreal	None	1	45.479984	-73.580070	Felinus	H4C 1

ETL- filter and clean the data from Yelp

CLEAN & TRANSFORM DATA

```
# Join the two DataFrame according to 'business_id'
df_yelp = pd.merge(df_biz, df_covid, on = 'business_id')
df_yelp.shape
```

```
(209795, 22)
```

```
# Filter out Las Vegas
df_lv = df_yelp[df_yelp.city == 'Las Vegas']
```

```
# Filter out just 'Restaurants' from 'categories' column
df_lv_food = df_lv[df_lv.categories.str.contains("Restaurants",na=False)]
```

```
# Drop the 'hours' columns
df_lv_food = df_lv_food.drop(columns = ['hours'])
```

```
# Filter out only restaurants that are open
df_lv_food = df_lv_food[df_lv_food.is_open == 1]
```

```
# Then drop the 'is_open' column
df_lv_food = df_lv_food.drop(columns = ['is_open'])
```

```
df_lv_food.shape
```

```
(4386, 20)
```

```
# Parsing a dictionary in a pandas dataframe cell into new row cells (new columns)
df_lv_food__attributes = df_lv_food.attributes.apply(pd.Series)
# Take only the relevant columns
df_lv_food__attributes_useful = df_lv_food__attributes[[
    'RestaurantsDelivery',
    'RestaurantsTakeOut',
    'RestaurantsReservations',
    'OutdoorSeating',
    'RestaurantsGoodForGroups',
    'DriveThru',
    'ByAppointmentOnly'
    ]]
df_lv_food__attributes_useful.head()
```

	RestaurantsDelivery	RestaurantsTakeOut	RestaurantsReservations	OutdoorSeating	RestaurantsGoodForGroups	DriveThru	ByAppointmentOnly
86	False	True	False	NaN	NaN	NaN	NaN
101	False	True	False	False	True	True	NaN
157	True	True	False	False	True	False	NaN
238	False	True	False	False	True	NaN	False
246	True	True	True	False	True	NaN	False

```
df = pd.concat([df_lv_food,df_lv_food__attributes_useful], axis = 1)
df = df.drop(columns = ['attributes'])
df
```

	address	business_id	categories	city	latitude	longitude	name	postal_code	review_count	stars	state	Call To Action enabled
86		OWkS1FXNJbozn-qPg3LWxg	Food, Food Trucks, Restaurants, Pizza	Las Vegas	36.128561	-115.171130	Mama Napoli Pizza	89109	11	4.5	NV	FALSE

ETL- extract data from Zomato


```
def get_location_details(query):

    headers = {
        'Accept': 'application/json',
        'user-key': zomato_api,
    }
    params = (
        ('query', query),
    )

    response = requests.get('https://developers.zomato.com/api/v2.1/locations', headers=headers, params=params)
    data = response.json()

    for loc in data['location_suggestions']:
        loc_id = loc['entity_id']
        loc_type = loc['entity_type']

    return loc_id, loc_type
```

```
def get_restaurants(ent_id, ent_type):

    headers = {
        'Accept': 'application/json',
        'user-key': zomato_api,
    }

    params = (
        ('entity_id', ent_id),
        ('entity_type', ent_type),
    )

    response = requests.get('https://developers.zomato.com/api/v2.1/search', headers=headers, params=params)

    return response.json()
```

	restaurant.name	restaurant.R.res_id	restaurant.location.address	restaurant.all_reviews_count	restaurant.user_rating.aggregate_rating	restaurant.user_rating.votes
0	Mon Ami Gabi	16979446	3655 Las Vegas Boulevard South 89109	319	4.6	1214
1	Gordon Ramsay Burger	16982906	3667 South Las Vegas Boulevard 89109	285	4.5	608
2	Bouchon Bistro - The Venetian	16977639	The Venetian, 3355 Las Vegas Boulevard South, ...	175	4.6	693
3	Lotus of Siam	16979203	953 E Sahara Ave, Las Vegas 89104	157	4.6	606
4	Mesa Grill	16979381	3570 South Las Vegas Boulevard 89109	222	4.3	812
5	The Buffet at Bellagio	16981060	3600 Las Vegas Blvd S, Las Vegas 89109	167	4.0	709
6	Burger Bar - Mandalay Bay	16977697	Mandalay Bay, 3930 South Las Vegas Boulevard, ...	147	4.2	715

ETL-Nevada COVID-19 Case Data

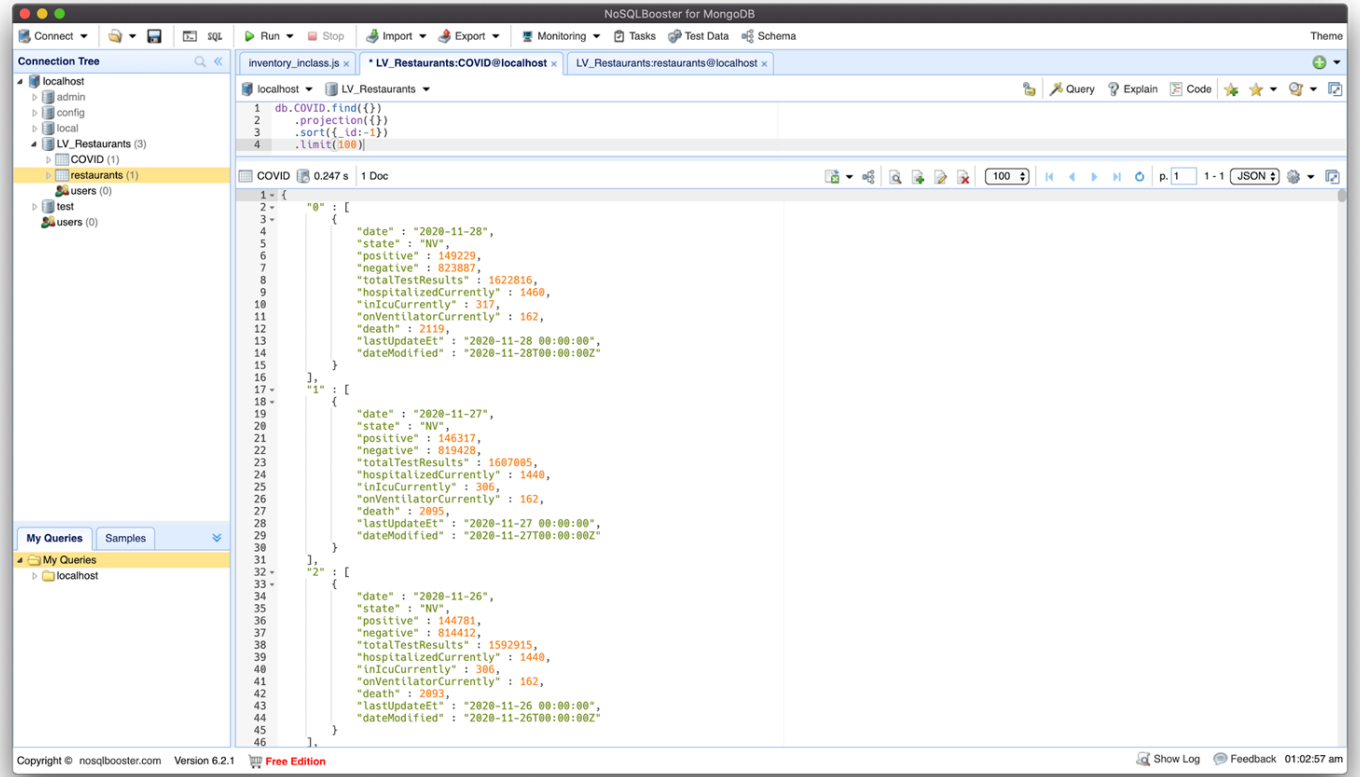

```
url="http://covidtracking.com/api/states/daily.csv"
s=requests.get(url).content
```

```
covid_df = pd.read_csv(io.StringIO(s.decode('utf-8')))
```

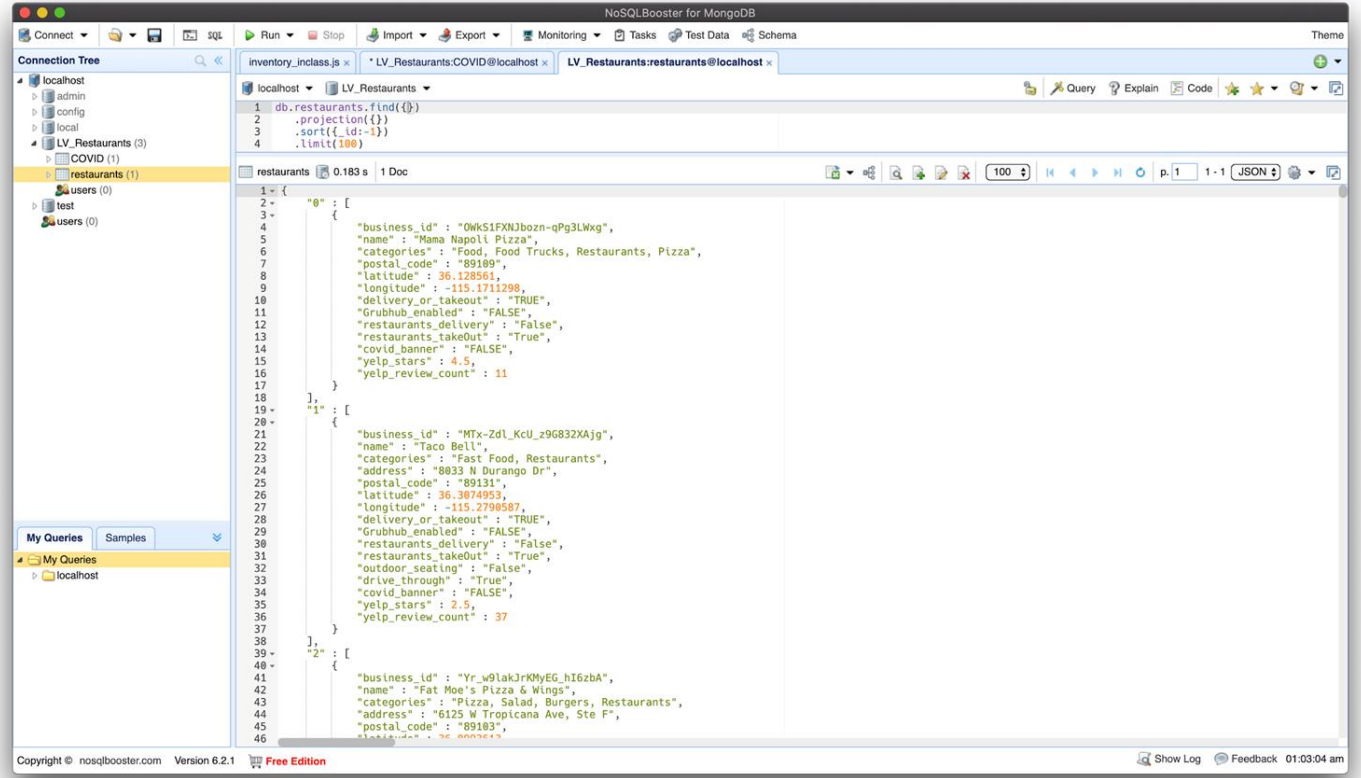
```
# format
covid_df = pd.read_csv(io.StringIO(s.decode('utf-8')))
covid_df['date'] =  pd.to_datetime(covid_df['date'], format='%Y%m%d')
covid_df.drop(['dateChecked'],axis=1,inplace=True)
covid_df['state']=covid_df['state'].apply(str)
covid_df.fillna(value=-1, inplace=True)
covid_df.info()
```

	date	state	positive	negative	totalTestResults	hospitalizedCurrently	inIcuCurrently	onVentilatorCurrently	recovered	death	lastUpdateEt	dateModified
36	2020-11-12	NV	114880.0	750483.0	1375669.0	941.0	229.0	115.0	2591.0	1880.0	11/12/2020 00:00	2020-11-12T00:00:00Z
92	2020-11-11	NV	113411.0	745908.0	1362884.0	950.0	233.0	107.0	2560.0	1877.0	11/11/2020 00:00	2020-11-11T00:00:00Z
148	2020-11-10	NV	112304.0	742230.0	1352746.0	898.0	231.0	101.0	2560.0	1859.0	11/10/2020 00:00	2020-11-10T00:00:00Z
204	2020-11-09	NV	110982.0	739959.0	1344308.0	891.0	254.0	100.0	2550.0	1852.0	11/9/2020 00:00	2020-11-09T00:00:00Z
260	2020-11-08	NV	110022.0	736226.0	1335632.0	826.0	206.0	89.0	2525.0	1851.0	11/8/2020 00:00	2020-11-08T00:00:00Z
316	2020-11-07	NV	108746.0	733258.0	1326378.0	826.0	206.0	89.0	2525.0	1850.0	11/7/2020 00:00	2020-11-07T00:00:00Z
372	2020-11-06	NV	106922.0	728492.0	1312492.0	770.0	183.0	85.0	2525.0	1845.0	11/6/2020 00:00	2020-11-06T00:00:00Z
428	2020-11-05	NV	105360.0	724213.0	1299121.0	740.0	170.0	75.0	2511.0	1824.0	11/5/2020 00:00	2020-11-05T00:00:00Z

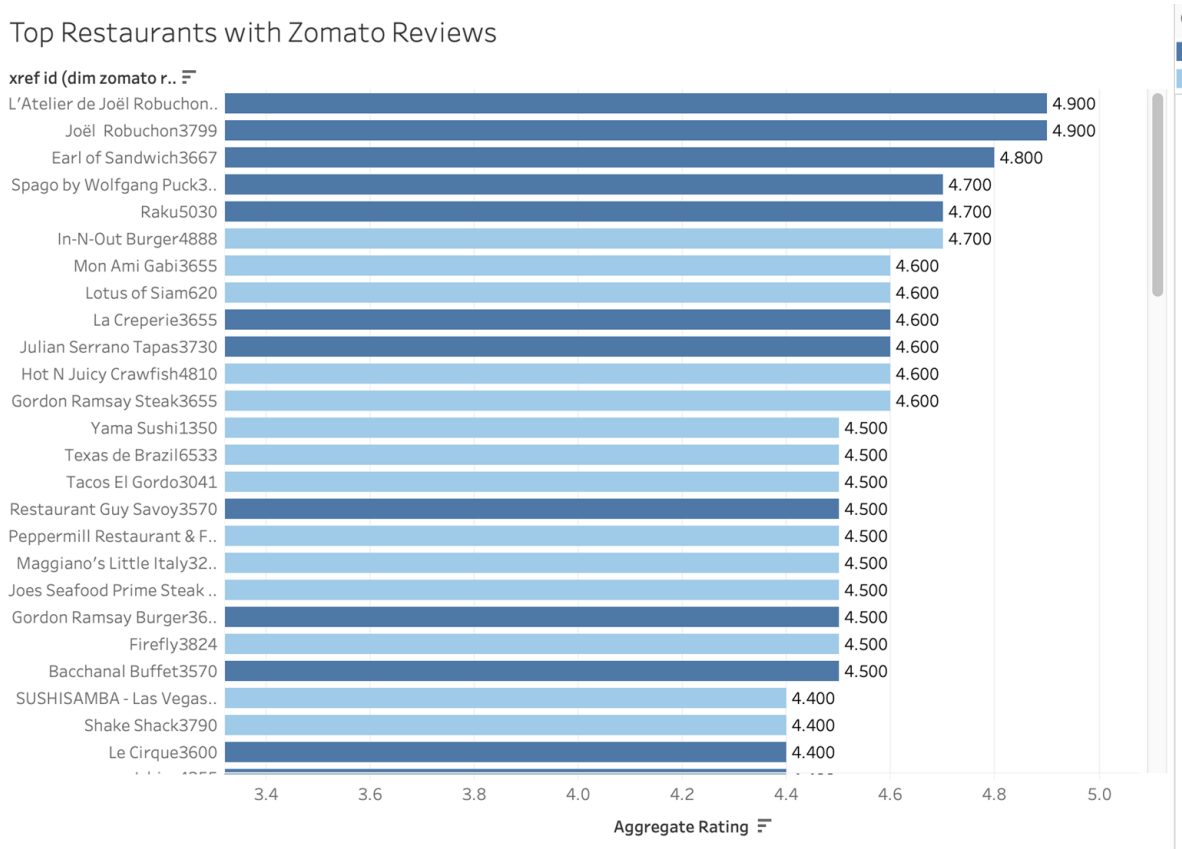
NoSQL Database - COVID data



NoSQL Database - restaurant data



05 Top Las Vegas Restaurants (Zomato Ratings)



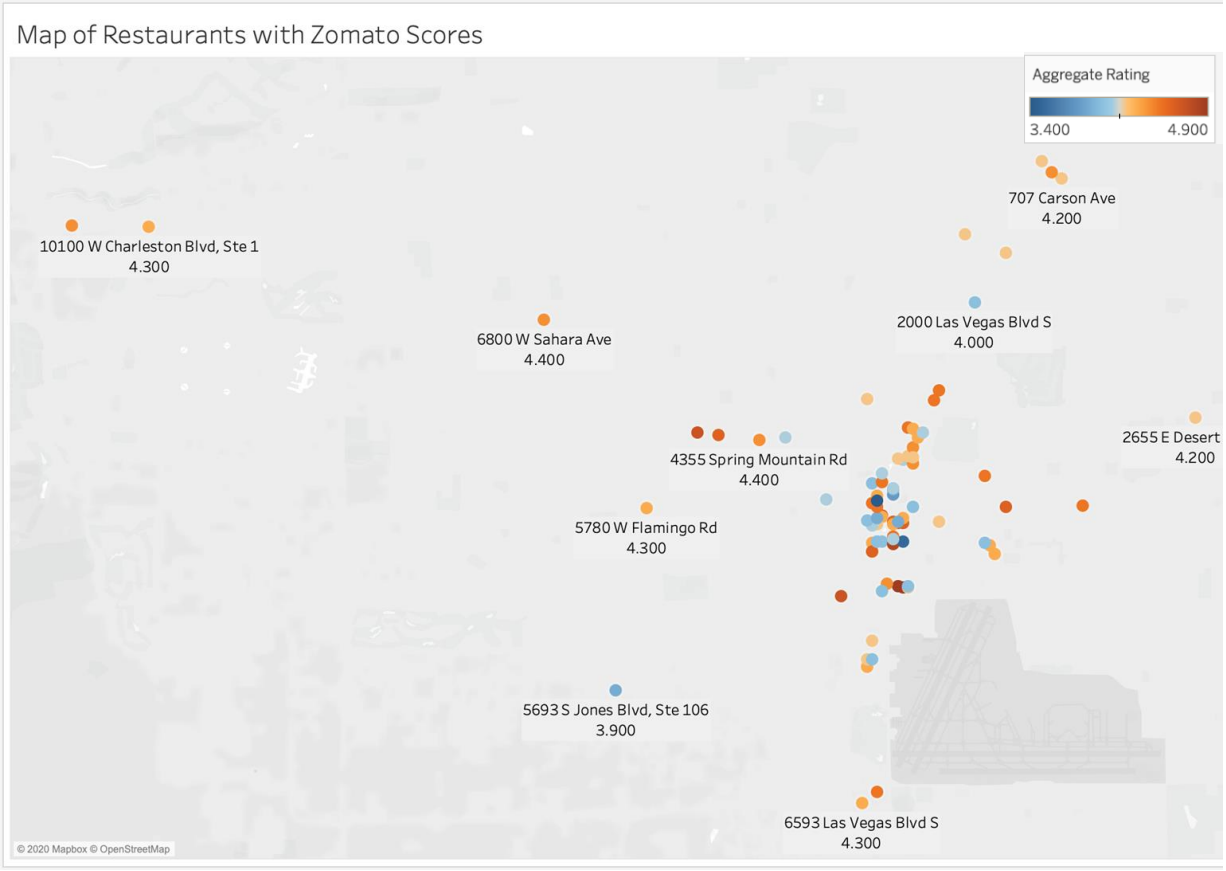
INSIGHT

The top three restaurants with high Zomato ratings are:

1. L'Atelier de Joel Robuchon
2. Joel Robuchon
3. Earl of Sandwich.

The top five Zomato rated restaurants are all not Grubhub enabled, however around half of the top 10 are Grubhub enabled.

05 Map of Restaurants with Zomato Scores



INSIGHT

Most restaurants with Zomato reviews are on the Las Vegas Strip, and the overall scores are high.