



# Manhattan Restaurant Rating



**Katz**

Katz School  
of Science and Health

Pratibha Bhalerao  
Yuchen Wang  
Xin Xiang  
Isaac Bendahan

# Problem

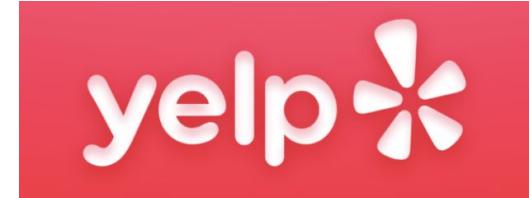
**How to choose restaurant based on cleanliness and rating?**

**Does a bad government health grading condemn a restaurant to failure?**

**Does a good government health grading ensure a good customer experience?**



# Solution



- We compared the government health grades of each restaurant with the rates on Yelp
- We created a rating analysis based on the votes of each customer
- We compared this rating analysis score to the health grades in order to see if they matched



# Data-Yelp(Unstructure Data)



Nicole K.

Added 1 photo

Fini Pizza



Andrea D.

Wrote a review



Santa Panza



Santa Panza is one of my favorite places in Bushwick. Everything on the menu is a hit. My...

[Continue reading](#)



Nicole K.

Wrote a review



Fini Pizza



One of the best slices in New York City. The hot honey is a must. Easy to grab a quick slice to go



Freeyah-Knycky C.

Added 1 photo

City Point BKLYN



Spencer K.

Added 2 photos

Ellington in the Park



Spencer K.

Wrote a review



Yeshiva University | Katz School

# Data-NYC OpenData(Structure Data)



Home Data About ▾ Learn ▾ Alerts Contact Us Blog |  Sign In

← Back to Primer    ↗ Switch to Grid View

↻ ↻  Search X

T CAMIS camis	T DBA dba	T BORO boro	T BUILDING building	T STREET street	T ZIPC zipco
50124829	DUNKIN	Manhattan	303	WEST 42 STREET	10036
50072352	WOK WOK RESTAURANT	Manhattan	11	MOTT STREET	10013
40868717	ALICE'S TEA CUP	Manhattan	102	WEST 73 STREET	10023
50051998	HUA XIA RESTAURANT	Manhattan	49	DIVISION STREET	10002
50002474	THE LOYAL	Manhattan	289	BLEECKER STREET	10014
10000707	DOUGHBOY	Manhattan	50	WEST 111 STREET	10000

< 1 of 743 >

Showing r...



Filters |  Clear all



Yeshiva University | Katz School

# Target Audience - Customers

## How to choose a restaurant?

Restaurant\_Selection by [Yuchen Wang](#)

Creator: Isaac Bendahan, Pratibha Bhalerao, Xin Xiang, Yuchen Wang

### Restaurant Selection

Cuisine	Name	Address	Grade	Rating
Asian/Asian Fusion	Raku	342 E 6th St, New York, NY 10003	A	4.5
	Sive Spice	227 Mulberry St, New York, NY 100..	A	4.5
	The Handpulled Noodle	3600 Broadway, Ste 3, New York, ..	A	4.0
	Real Kung Fu Little Steamed ..	811 8th Ave, New York, NY 10019	B	4.0
	Hanoi House	119 Saint Marks Pl, New York, NY 1...	A	4.0
	TAO Uptown	42 E 58th St, New York, NY 10022	B	3.5
Chinese	Wah Fung No 1	79 Chrystie St, New York, NY 10002	A	4.5
	Uluh	152 2nd Ave, Ste A, New York, NY 1..	A	4.5
	Szechuan Mountain House	23 St Marks Pl, New York, NY 10003	A	4.5
	Da Long Yi Hot Pot	159 Canal St, Fl 2 Unit E, New York, ..	C	4.5
	Vanessa's Dumpling House	118A Eldridge St, New York, NY 10..	A	4.0
	Taiwan Pork Chop House	3 Doyers St, New York, NY 10013	A	4.0
	Little Alley	550 3rd Ave, New York, NY 10016	B	4.0
	Kong Sihk Tong 港食堂	65 Bayard St, New York, NY 10013	C	4.0
	Hometown Hotpot & BBQ	194 Grand St, New York, NY 10013	A	4.0
	Han Dynasty	215 W 85th St, New York, NY 10024	C	4.0
	Dim Sum Palace	334 W 46th St, New York, NY 10036	A	4.0
	Japanese	Kame	330 8th Ave, New York, NY 10001	A
Benemon		108 E 4th St, New York, NY 10003	B	4.5
Tsuru Ton Tan		21 E 16th St, New York, NY 10003	C	4.0
Sushi Yasaka		251 W 72nd St, New York, NY 10023	A	4.0

Selected Restaurants Distribution

© Mapbox © OSM

102 St Marks Pl  
New York  
[View on Google Maps](#)



# Target Audience - Government Officer

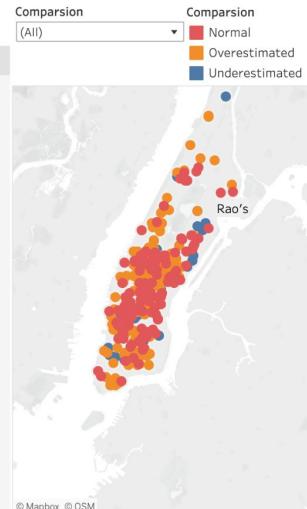
Do Grade and ratings match?

Restaurants Grade & Rating Comparsion

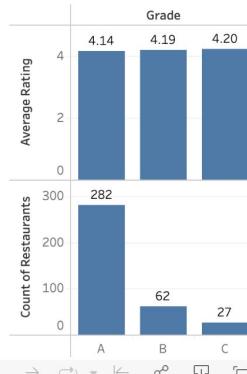
Grade & Rating Comparsion

Name	F	Grade	Rating Level	Comparsion	Rating
The Den		B	A	Underestimated	5.0
OOHU		A	A	Normal	5.0
Kati Shop		A	A	Normal	5.0
GAUDIr		A	A	Normal	5.0
Chamoun's Way		B	A	Underestimated	5.0
Aunt Bernie's Bar..		A	A	Normal	5.0
Zoralie Restaurant		A	A	Normal	4.5
YUBU		A	A	Normal	4.5
Wolf at Nordstro..		A	A	Normal	4.5
Wah Fung No 1		A	A	Normal	4.5
Valla Table		A	A	Normal	4.5
Valerie		B	A	Underestimated	4.5
Up Thai		A	A	Normal	4.5
Uluh		A	A	Normal	4.5
TSQ Food Court		B	A	Underestimated	4.5
Towa		A	A	Normal	4.5
Torrisi		A	A	Normal	4.5
TomoTomo		A	A	Normal	4.5
Three Roosters		B	A	Underestimated	4.5
The Tang - Upper ..		A	A	Normal	4.5
The Rag Trader & ..		A	A	Normal	4.5
The Lavaux		B	A	Underestimated	4.5
The Flatiron Roo..		A	A	Normal	4.5

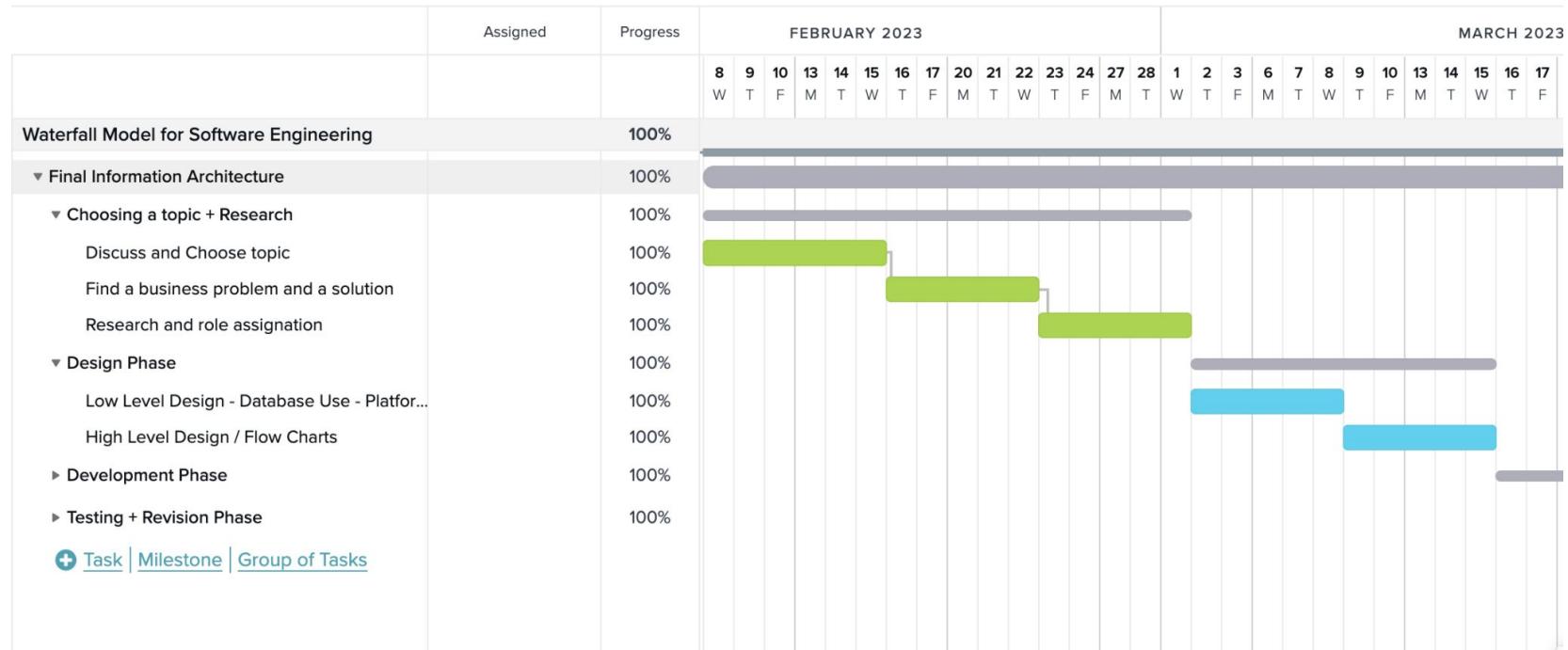
↑↑↑ + a b | e a u



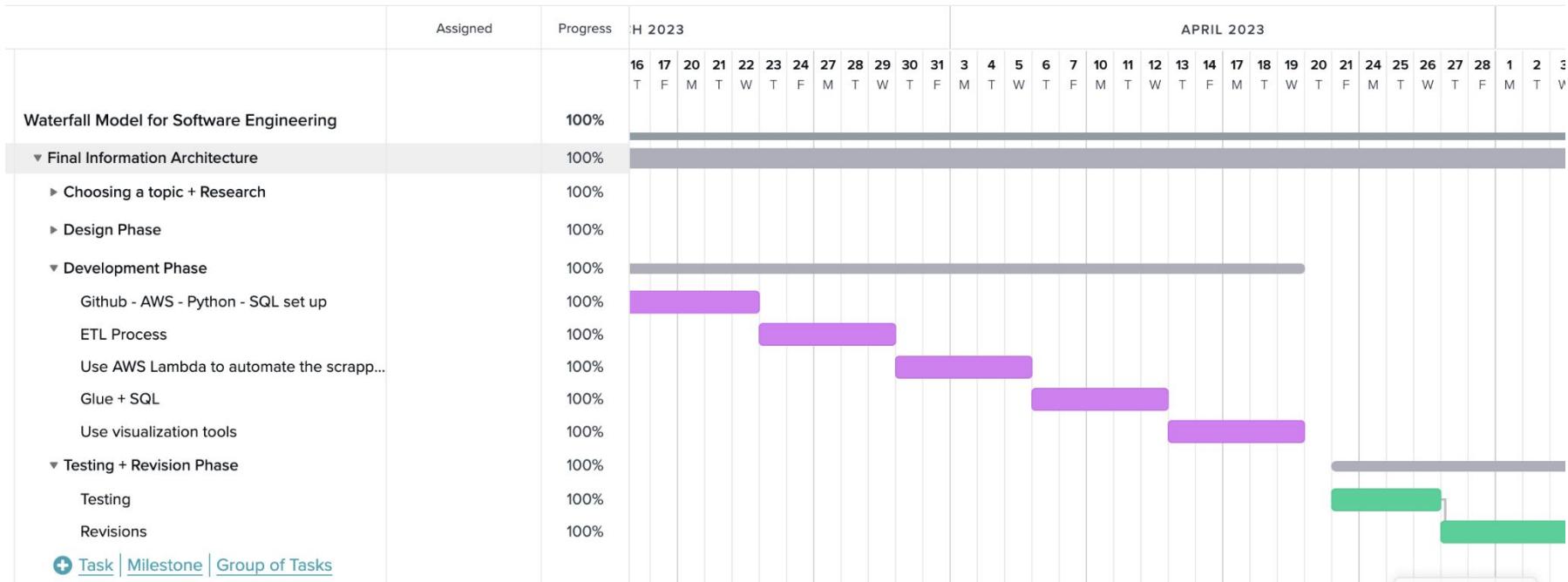
Grade VS. Rating  
&  
Count of Grade



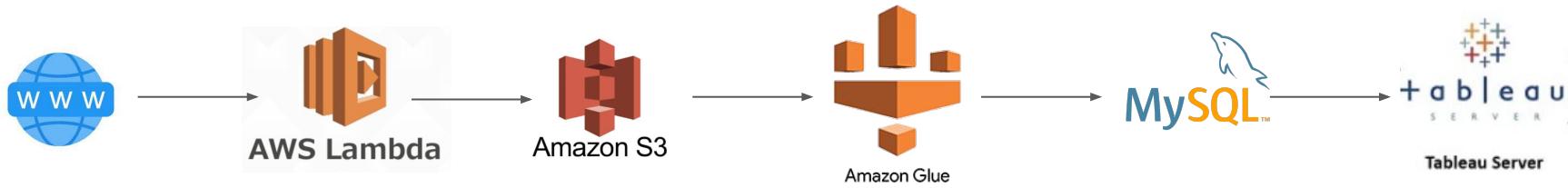
# Waterfall Project Management - Part 1



# Waterfall Project Management - Part 2



# AWS Solution Architecture Diagram



# Python file

## Scrape Yelp data

```
In [3]: import pandas as pd
import requests
pd.set_option('display.max_columns', None)
```

```
In [4]: # Yelp's API Data
import requests

api_key = '36x_mqeUCvrVSDBUthEGvXwYrh3JQZ81KTqJjBej-fSb59kO3JryRCqJG_NEPyAQq3bSaMZXurlYwhJCKwzp
headers = {'Authorization': 'Bearer %s' % api_key}

url = 'https://api.yelp.com/v3/businesses/search'

params = {
    'term': 'restaurants',
    'location': 'Manhattan, NY',
    'categories': 'restaurants',
    'limit': 50,
```



# Python file

## Scrape NYC Open data

```
In [8]: # NYC Open Data
url = "https://data.cityofnewyork.us/resource/43nn-pn8j.json"

# Set a limit for the number of rows to fetch in each request
limit = 20000

# Initialize an empty list to store the data
all_data = []

# Start with an offset of 0
offset = 0

while True:
    # Add the limit and offset parameters to the API request
    response = requests.get(url, params={'$limit': limit, '$offset': offset})

    if response.status_code == 200:
```



# Python file

**Load S3 Data in Python and upload the file to new created S3 Bucket in AWS**

```
In [27]: import logging
import boto3
from botocore.exceptions import ClientError
```

```
In [25]: import boto3

# Set AWS access and secret keys
ACCESS_KEY = 'AKIAWAI2HSUU5BKNTA2P'
SECRET_KEY = 'wGAY2d9AFZ8HAGd5gN0NLIIQwUjtcHWE01zmbzfm'

# Create a connection to your AWS account
s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY, aws_secret_access_key=SECRET_KEY)
```

```
In [40]: from datetime import datetime
```



# Modify Python to Lambda Format

```
import io
import os
import json
import pandas as pd
import requests
import boto3
from botocore.exceptions import ClientError

pd.set_option('display.max_columns', None)

# Yelp Data
api_key = os.environ['YELP_API_KEY']
url_y = 'https://api.yelp.com/v3/businesses/search'
# NYC Open Data
url_n = "https://data.cityofnewyork.us/resource/43nn-pn8j.json"

# Set AWS access and secret keys
ACCESS_KEY = os.environ['AWS_ACCESS_KEY']
SECRET_KEY = os.environ['AWS_SECRET_KEY']
bucket_name = "finalpiplinedata"

def save_to_s3(dataframe, bucket_name, file_name):
    csv_string = dataframe.to_csv(index=False)
    # Create a connection to your AWS account
    s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY, aws_secret_access_key=SECRET_KEY)
    # Upload files to the bucket
    s3_resource = boto3.resource('s3', aws_access_key_id=ACCESS_KEY, aws_secret_access_key=SECRET_KEY)
    bucket = s3_resource.Bucket(bucket_name)
    bucket.put_object(Key=file_name, Body=csv_string.encode())

def get_yelp_data(url_y):
    all_restaurants = []
    headers = {'Authorization': 'Bearer %s' % api_key}
    params = {
```



# Modify Python to Lambda Format

```
# Loop through the maximum number of pages (1000 results / 50 results per page)
for offset in range(0, 1000, 50):
    params['offset'] = offset
    response = requests.get(url_y, headers=headers, params=params)

    if response.status_code == 200:
        data = response.json()
        all_restaurants.extend(data['businesses'])
    else:
        print("Error: Unable to retrieve data")
        break

yelp = pd.json_normalize(all_restaurants)
#Keep 9 digital numbers only.
yelp['phone'] = yelp['phone'].str.slice(2)
yelp_data=yelp
return yelp_data

def get_nyc_open_data(url_n):
    # Set a limit for the number of rows to fetch in each request
    limit = 20000

    # Initialize an empty list to store the data
    all_data = []

    # Start with an offset of 0
    offset = 0

    while True:
        # Add the limit and offset parameters to the API request
        response = requests.get(url_n, params={'$limit': limit, '$offset': offset})

        if response.status_code == 200:
            data = response.json()

            # Check if there's any data left to fetch
            if not data:
                break
```



# Modify Python to Lambda Format

```
# Add the fetched data to the list
all_data.extend(data)

# Update the offset for the next request
offset += limit
else:
    print(f"Failed to fetch data (status code: {response.status_code})")
    break
nyc = pd.DataFrame(all_data)
# Only focus on Manhattan restaurants
nyc_data = nyc.loc[nyc['boro']=='Manhattan']
return nyc_data

def lambda_handler(event, context):
    yelp_data = get_yelp_data(url_y)
    nyc_open_data = get_nyc_open_data(url_n)

    save_to_s3(yelp_data, "yelp_data.csv")
    save_to_s3(nyc_open_data, "nyc_open_data.csv")

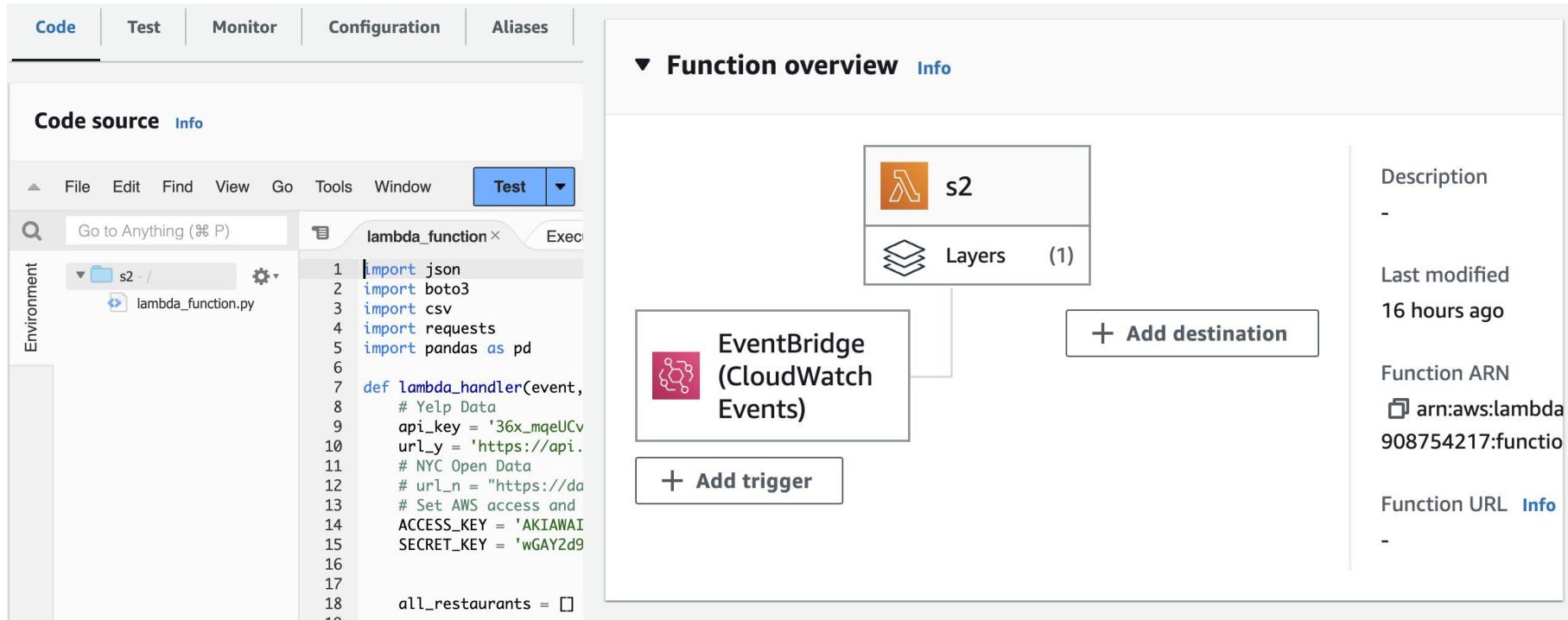
    return {
        "statusCode": 200,
        "body": json.dumps({"message": "Data saved to S3"})
    }

#creat file named project and a subfile called package
#open terminal

#cd path-to-project
#virtualenv venv
#source venv/bin/activate
#pip install pandas requests boto3
```



# Lambda



The image shows a split-screen view of the AWS Lambda console and a code editor.

**Code Editor (Left):**

- Header:** Code, Test, Monitor, Configuration, Aliases.
- Section:** Code source, Info.
- Toolbar:** File, Edit, Find, View, Go, Tools, Window, Test (highlighted).
- Search:** Go to Anything (% P).
- Environment:** s2 / (selected), lambda\_function.py.
- Code:**

```
1 import json
2 import boto3
3 import csv
4 import requests
5 import pandas as pd
6
7 def lambda_handler(event,
8     # Yelp Data
9     api_key = '36x_mqeUCv
10    url_y = 'https://api.
11    # NYC Open Data
12    # url_n = "https://da
13    # Set AWS access and
14    ACCESS_KEY = 'AKIAWAI
15    SECRET_KEY = 'wGAY2d9
16
17
18    all_restaurants = []
```

**Function Overview (Right):**

- Header:** Function overview, Info.
- Function Details:** s2, Layers (1).
- Triggers:** EventBridge (CloudWatch Events) (selected), + Add trigger.
- Destinations:** + Add destination.
- Table:**

Description
-
Last modified
16 hours ago
Function ARN
arn:aws:lambda:908754217:function
Function URL
-



# Lambda

## Trigger



### EventBridge (CloudWatch Events): [Import\\_data\\_every\\_week](#)

arn:aws:events:us-east-1:412908754217:rule/Import\_data\_every\_week

Rule state: **ENABLED**

#### ▼ Details

Event bus: **default**

name: **Import\_data\_every\_week**

Schedule expression: **cron(0 12 ? \* MON \*)**

Service principal: **events.amazonaws.com**

Statement ID: **lambda-fabb717a-584f-4b87-9e6a-326f33e2acd6**

url: [events/home#/rules/Import\\_data\\_every\\_week](#)



# S3

A screenshot of a Microsoft Excel spreadsheet titled "yelp\_data". The spreadsheet has a green header bar with various icons and buttons. Below the header, there's a toolbar with "View", "Zoom", "Add Category", "Pivot Table", "Insert", "Table", "Chart", "Text", "Shape", "Media", "Comment", and "Col". A green button labeled "Sheet 1" is visible. The main content area shows a table with several columns: camis, boro, building, street, zipcode, phone, inspection\_date, critical\_flag, and re. The data consists of ten rows, each containing a unique camis number, the borough (Manhattan), the building number, the street name, the zip code, the phone number, the inspection date (all set to 1900-01-01 or 2023-03-04), the critical flag (all set to "Not Applicable"), and a redacted value.

camis	boro	building	street	zipcode	phone	inspection_date	critical_flag	re
50129207	Manhattan	920	2 AVENUE	10017	1914258258	1900-01-01T00:00:00.000	Not Applicable	21
50044593	Manhattan	4	PENN PLZ	10121	2124656273	2023-03-04T00:00:00.000	Not Applicable	21
50120018	Manhattan	18	WEST 23 STREET	10010	3052064247	1900-01-01T00:00:00.000	Not Applicable	21
50107930	Manhattan	416	BROADWAY	10013	646 4550323	1900-01-01T00:00:00.000	Not Applicable	21
50128013	Manhattan	225	WEST 35 STREET	10001	6466961689	1900-01-01T00:00:00.000	Not Applicable	21
50129852	Manhattan	708	3 AVENUE	10017	6782210550	1900-01-01T00:00:00.000	Not Applicable	21
50130647	Manhattan	124	EAST 14 STREET	10003	9177566709	1900-01-01T00:00:00.000	Not Applicable	21
50135844	Manhattan	138	5 AVENUE	10011	6094392777	1900-01-01T00:00:00.000	Not Applicable	21
50130450	Manhattan	203	CHRYSTIE STREET	10002	3472628409	1900-01-01T00:00:00.000	Not Applicable	21
50119979	Manhattan	2656	BROADWAY	10025	6178200088	1900-01-01T00:00:00.000	Not Applicable	21



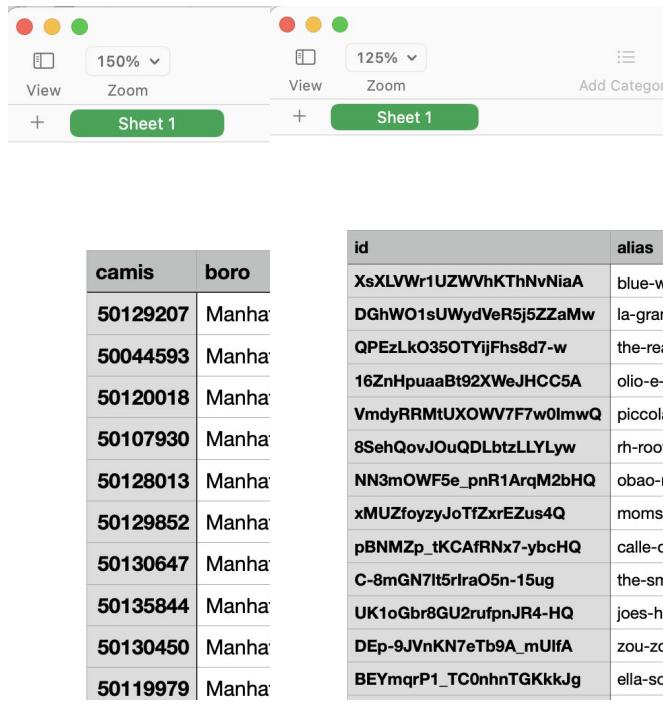
# S3

The screenshot shows a spreadsheet application window with the title bar 'nyc\_data'. The menu bar includes 'View', 'Zoom', 'Add Category', 'Pivot Table', 'Insert', 'Table', 'Chart', 'Text', 'Shape', 'Media', 'Comment', and 'Coll'. The toolbar has buttons for View (150%, 125%), Zoom, Insert, Table, Chart, Text, Shape, Media, Comment, and Coll. There are two tabs: 'Sheet 1' (selected) and another 'Sheet 1'. The table has columns: camis, boro, id, alias, name, and image\_url.

camis	boro	id	alias	name	image_url
50129207	Manhattan	XsXLVWr1UZWVhKThNvNiaA	blue-willow-夜来湘-new-york-2	Blue Willow 夜来湘	<a href="https://s3-media2.fl.yelpcdn.com">https://s3-media2.fl.yelpcdn.com</a>
50044593	Manhattan	DGhWO1sUWydVeR5j5ZZaMw	la-grande-boucherie-new-york-2	La Grande Boucherie	<a href="https://s3-media3.fl.yelpcdn.com">https://s3-media3.fl.yelpcdn.com</a>
50120018	Manhattan	QPEzLkO35OTYijFhs8d7-w	the-reading-room-new-york	The Reading Room	<a href="https://s3-media2.fl.yelpcdn.com">https://s3-media2.fl.yelpcdn.com</a>
50107930	Manhattan	16ZnHpuuaBt92XWeJHCC5A	olio-e-più-new-york-7	Olio e Più	<a href="https://s3-media3.fl.yelpcdn.com">https://s3-media3.fl.yelpcdn.com</a>
50128013	Manhattan	VmdyRRMtUXOWV7F7w0ImwQ	piccola-cucina-uptown-new-york	Piccola Cucina Uptown	<a href="https://s3-media1.fl.yelpcdn.com">https://s3-media1.fl.yelpcdn.com</a>
50129852	Manhattan	8SehQovJOUqdLbtzLLYLyw	rh-rooftop-restaurant-new-york-new-york-2	RH Rooftop Restaurant New York	<a href="https://s3-media3.fl.yelpcdn.com">https://s3-media3.fl.yelpcdn.com</a>
50130647	Manhattan	NN3mOWF5e_pnR1ArqM2bHQ	obao-new-york-3	OBAO	<a href="https://s3-media4.fl.yelpcdn.com">https://s3-media4.fl.yelpcdn.com</a>
50135844	Manhattan	xMUZfoyzyJoTfZxrEZus4Q	moms-kitchen-and-bar-new-york	Mom's Kitchen & Bar	<a href="https://s3-media3.fl.yelpcdn.com">https://s3-media3.fl.yelpcdn.com</a>
50130450	Manhattan	pBNMZh_tKCAFRNx7-ybcHQ	calle-dao-chelsea-new-york-3	Calle Dao Chelsea	<a href="https://s3-media1.fl.yelpcdn.com">https://s3-media1.fl.yelpcdn.com</a>
50119979	Manhattan	C-8mGN7It5rlraO5n-15ug	the-smith-new-york-2	The Smith	<a href="https://s3-media2.fl.yelpcdn.com">https://s3-media2.fl.yelpcdn.com</a>
		UK1oGbr8GU2rufpnJR4-HQ	joes-home-of-soup-dumplings-new-york	Joe's Home of Soup Dumplings	<a href="https://s3-media2.fl.yelpcdn.com">https://s3-media2.fl.yelpcdn.com</a>
		DEp-9JVnKN7eTb9A_mUlfa	zou-zou-s-new-york-3	Zou Zou's	<a href="https://s3-media1.fl.yelpcdn.com">https://s3-media1.fl.yelpcdn.com</a>
		BEYmqrP1_TC0nhnTGKkkJg	ella-social-new-york	Ella Social	<a href="https://s3-media1.fl.yelpcdn.com">https://s3-media1.fl.yelpcdn.com</a>



# S3



The image shows two Google Sheets side-by-side. Both sheets have a header row with columns labeled 'camis' and 'boro'. The first sheet contains 10 rows of data, while the second sheet has 4 rows. The data appears to be a subset of the first sheet's data.

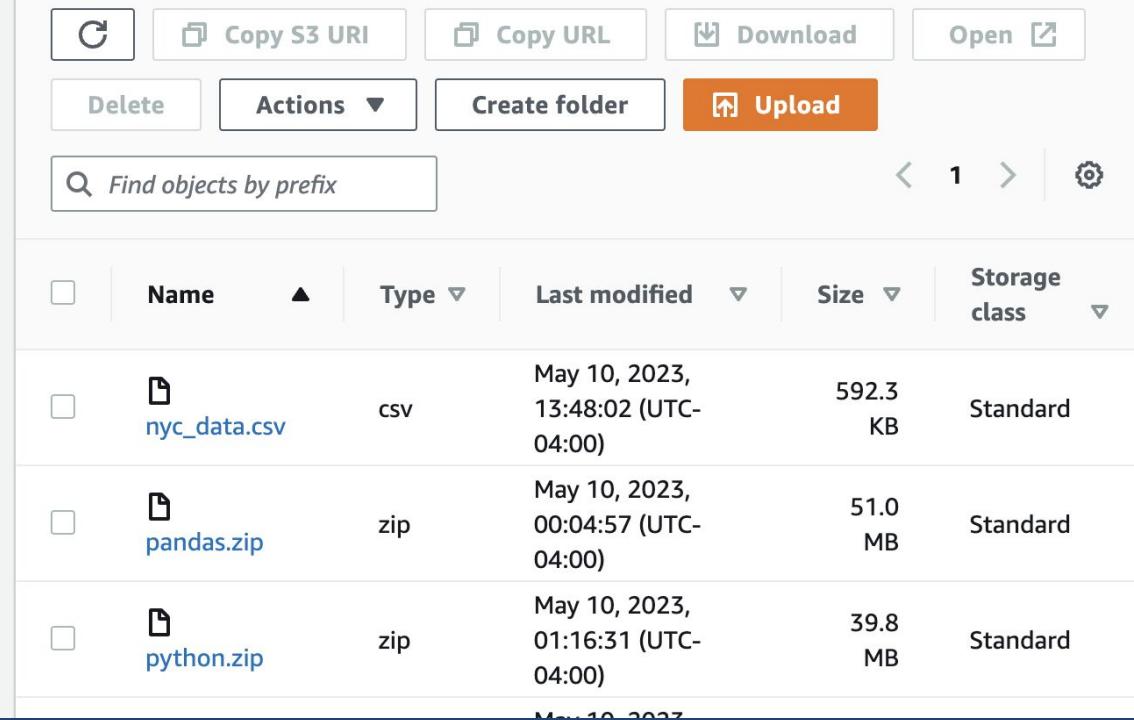
camis	boro
50129207	Manha
50044593	Manha
50120018	Manha
50107930	Manha
50128013	Manha
50129852	Manha
50130647	Manha
50135844	Manha
50130450	Manha
50119979	Manha

id	alias
XsXLVWr1UZWvhKThNvNiaA	blue-wi
DGhWO1sUWydVeR5j5ZZaMw	la-gran
QPEzLkO35OTYijFhs8d7-w	the-re
16ZnHpuuaBt92XWeJHCC5A	olio-e-
VmdyRRMtUXOWV7F7w0lmwQ	piccola
8SehQovJOUqDlbzLLYLyw	rh-roofl
NN3mOWF5e_pnR1ArqM2bHQ	obao-n
xMUZfozyJoTfZxrEZus4Q	moms-
pBNMZp_tKCafRNx7-ybcHQ	calle-d
C-8mGN7lt5rlraO5n-15ug	the-sm
UK1oGbr8GU2rufpnJR4-HQ	joes-hc
DEp-9JVnKN7eTb9A_mUlfa	zou-zou
BEYmqrP1_TC0nhnTGKkjg	ella-so

## Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



The screenshot shows the AWS S3 'Objects' list interface. At the top, there are several buttons: 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. Below these are buttons for 'Find objects by prefix' and navigation arrows. The main area displays a table of objects with columns: Name, Type, Last modified, Size, and Storage class. There are four objects listed:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">nyc_data.csv</a>	CSV	May 10, 2023, 13:48:02 (UTC-04:00)	592.3 KB	Standard
<input type="checkbox"/>	<a href="#">pandas.zip</a>	ZIP	May 10, 2023, 00:04:57 (UTC-04:00)	51.0 MB	Standard
<input type="checkbox"/>	<a href="#">python.zip</a>	ZIP	May 10, 2023, 01:16:31 (UTC-04:00)	39.8 MB	Standard



# MySQL Database

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view displays the 'innodb' schema and the 'restaurant' schema, which contains a 'Tables' folder with a selected 'restaurant\_reviews' table. Other objects in the 'restaurant' schema include 'Views', 'Stored Procedures', and 'Functions'. Below the schema tree, the 'restaurant\_starsschema' schema is also listed with its own 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main workspace on the right shows the SQL code for creating the 'restaurant\_reviews' table:

```
1 • CREATE TABLE restaurant_reviews (
2     name VARCHAR(255),
3     review_count INT,
4     latitude DECIMAL(10, 8),
5     longitude DECIMAL(11, 8),
6     zipcode VARCHAR(10),
7     grade_date DATE,
8     cuisine_description VARCHAR(255),
9     score INT,
10    grade CHAR(1),
11    rating DECIMAL(3, 2)
12 );
13
```



# Glue Crawler

AWS Glue > Databases

## Databases (2)

A database is a set of associated table definitions, organized into a logical group.

Last updated (UTC)  
May 10, 2023 at 21:46:00



Edit

Delete

Add database

Filter databases

< 1 >

<input type="checkbox"/>	Name	▲	Descrip...	▼	Locatio...	▼	Created on (UTC)	▼
<input type="checkbox"/>	irs990	-	jdbc:mysql://dat	May 4, 2023 at 22:12:59				
<input type="checkbox"/>	restaurant	-	jdbc:mysql://res	May 7, 2023 at 17:27:39				

## Tables (1)

View and manage all available tables.

Last updated (UTC)  
May 10, 2023 at 21:46:48



Delete

Data quality New

Add tables using crawler

Add table

Filter tables

< 1 >

✓ Crawler successfully starting  
The following crawler is now starting: "crawler-restaurant"

AWS Glue > Crawlers

## Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

### Crawlers (1/3)

View and manage all available crawlers.

Last updated (UTC)  
May 10, 2023 at 21:45:12



Action

Run

Create crawler

Filter crawlers

< 1 >

<input type="checkbox"/>	Name	State	Schedule
<input type="checkbox"/>	crawler	Ready	
<input type="checkbox"/>	crawler-restaurant	Ready	



# Join Tables

## Yelp Data

camis  
boro  
building  
street  
zipcode  
phone  
inspection\_date  
critical\_flag  
record\_date  
dba  
latitude  
longitude  
community\_board  
council\_district  
census\_tract  
bin  
bbi  
nta  
cuisine\_description  
action  
inspection\_type  
score  
grade  
grade\_date  
violation\_code  
violation\_description

## NYC Open Data

id  
alias  
name  
image\_url  
is\_closed  
url  
review\_count  
categories  
rating  
transactions  
phone  
display\_phone  
distance  
coordinates.latitude  
coordinates.longitude  
location.address1  
location.address2  
location.address3  
location.city  
location.zip\_code  
location.country  
location.state  
location.display\_address  
price

## Final Data

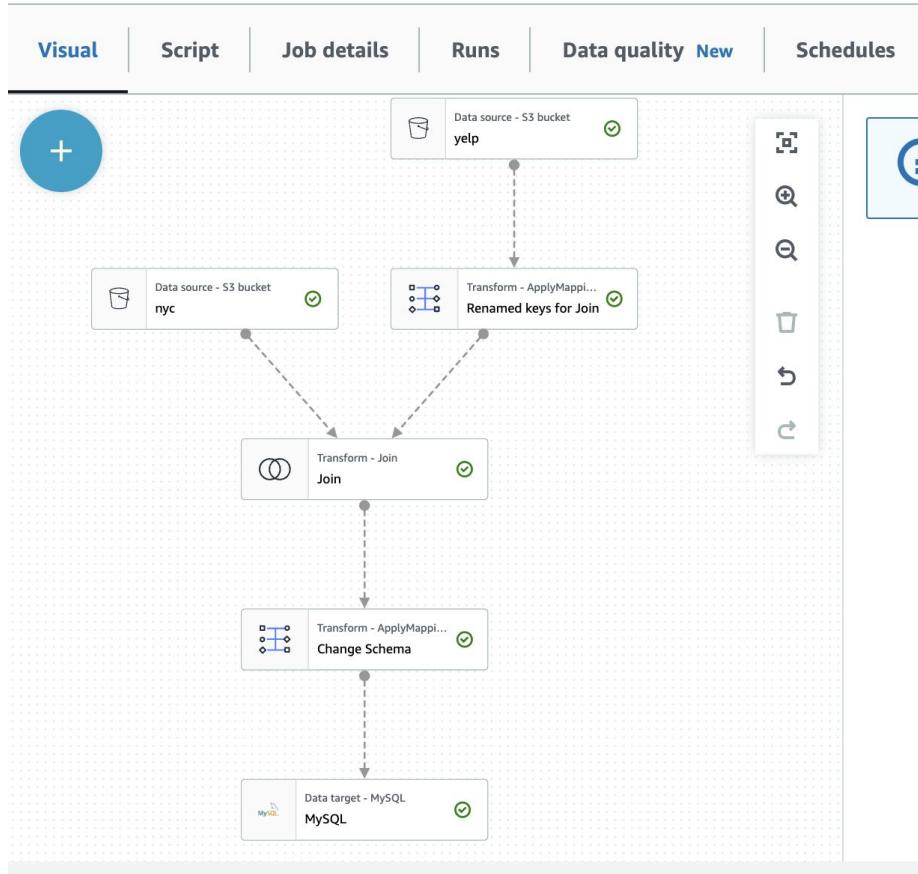
**'name',  
'review\_count',  
'location.display\_address',  
'latitude',  
'longitude',  
'zipcode',  
'grade\_date',  
'cuisine\_descripti  
on',  
'score',  
**'Grade'**,  
'rating'**



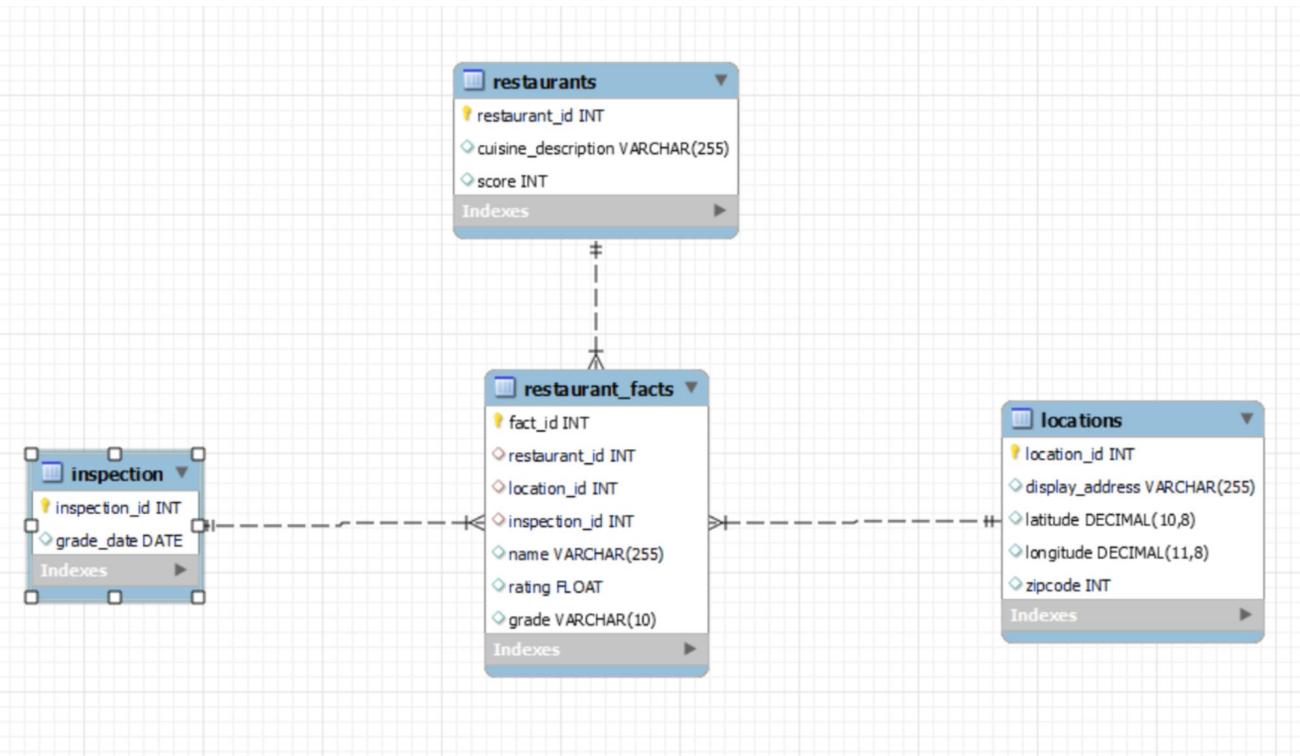
## ETL job

Last modified

# Glue Job

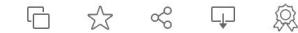


# MySQL Star Schema



# Tableau

Restaurant\_Selection by [Yuchen Wang](#)



## Restaurant Selection

Cuisine      Review CNT Group  
(Multiple values)      (Multiple values)

Cuisine	Name	Address	Grade	Rating
Asian/Asian Fusion	Raku	342 E 6th St, New York, NY 10003	A	4.5
	Sive Spice	227 Mulberry St, New York, NY 100..	A	4.5
	The Handpulled Noodle	3600 Broadway, Ste 3, New York, N..	A	4.0
	Real Kung Fu Little Steamed ..	811 8th Ave, New York, NY 10019	B	4.0
	Hanoi House	119 Saint Marks Pl, New York, NY 1..	A	4.0
	TAO Uptown	42 E 58th St, New York, NY 10022	B	3.5
Chinese	Wah Fung No 1	79 Chrystie St, New York, NY 10002	A	4.5
	Uluh	152 2nd Ave, Ste A, New York, NY 1..	A	4.5
	Szechuan Mountain House	23 St Marks Pl, New York, NY 10003	A	4.5
	Da Long Yi Hot Pot	159 Canal St, Fl 2 Unit E, New York, ..	C	4.5
	Vanessa's Dumpling House	118A Eldridge St, New York, NY 10..	A	4.0
	Taiwan Pork Chop House	3 Doyers St, New York, NY 10013	A	4.0
	Little Alley	550 3rd Ave, New York, NY 10016	B	4.0
	Kong Sihl Tong 港食堂	65 Bayard St, New York, NY 10013	C	4.0
	Hometown Hotpot & BBQ	194 Grand St, New York, NY 10013	A	4.0
	Han Dynasty	215 W 85th St, New York, NY 10024	C	4.0
Japanese	Dim Sum Palace	334 W 46th St, New York, NY 10036	A	4.0
	Kame	330 8th Ave, New York, NY 10001	A	4.5
	Benemon	108 E 4th St, New York, NY 10003	B	4.5
	Tsuru Ton Tan	21 E 16th St, New York, NY 10003	C	4.0

Creator:  
Isaac Bendahan,  
Pratibaha Bhalerao,  
Xin Xiang,  
Yuchen Wang

### Selected Restaurants Distribution



[https://public.tableau.com/app/profile/lawrence.wang2015/viz/Restaurant\\_Selection/Dashboard1](https://public.tableau.com/app/profile/lawrence.wang2015/viz/Restaurant_Selection/Dashboard1)



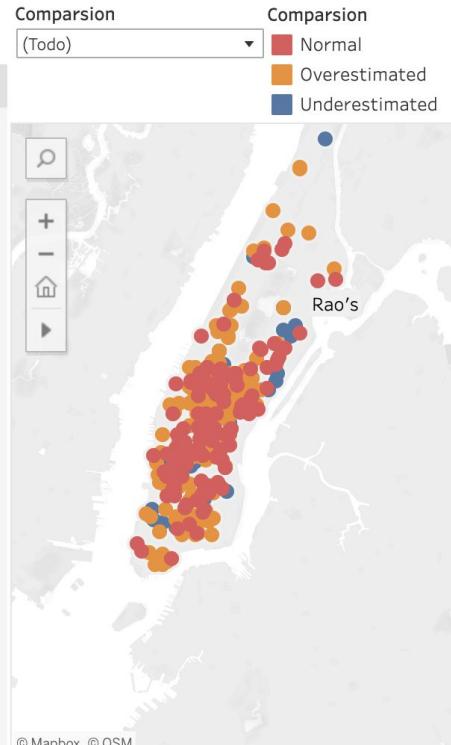
Yeshiva University | Katz School

# Restaurants Grade & Rating Comparsion

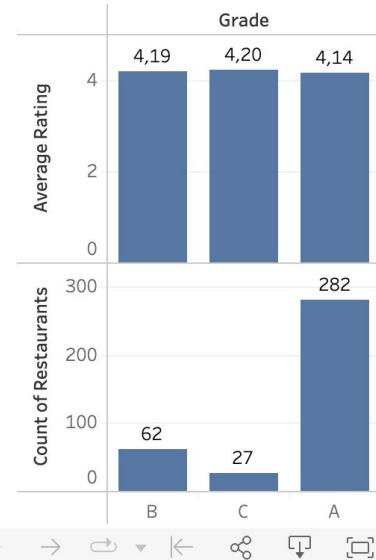
## Grade & Rating Comparsion

Name	F	Grade	Rating Level	Comparsion	Rating
The Den		B	A	Underestimated	5,0
OOHU		A	A	Normal	5,0
Kati Shop		A	A	Normal	5,0
GAUDIr		A	A	Normal	5,0
Chamoun's Way		B	A	Underestimated	5,0
Aunt Bernie's Bar..		A	A	Normal	5,0
Zoralie Restaurant		A	A	Normal	4,5
YUBU		A	A	Normal	4,5
Wolf at Nordstro..		A	A	Normal	4,5
Wah Fung No 1		A	A	Normal	4,5
Valla Table		A	A	Normal	4,5
Valerie		B	A	Underestimated	4,5
Up Thai		A	A	Normal	4,5
Uluh		A	A	Normal	4,5
TSQ Food Court		B	A	Underestimated	4,5
Towa		A	A	Normal	4,5
Torrisi		A	A	Normal	4,5
TomoTomo		A	A	Normal	4,5
Three Roosters		B	A	Underestimated	4,5
The Tang - Upper ..		A	A	Normal	4,5
The Rag Trader & ..		A	A	Normal	4,5
The Lavaux		B	A	Underestimated	4,5
The Flatiron Roo..		A	A	Normal	4,5

++ + a b | e a u



## Grade VS. Rating & Count of Grade



<https://public.tableau.com/app/profile/lawrencewang2015/viz/RestaurantGradeRatingComparsion/Dashboard2?publish=yes>



# Challenges



- Reduced and limited access to Yelp API
- Matching the Yelp ratings to the NYC Open Data
- Building the ELT using Glue - hard to find the problem in the job of AWS Glue
- Upload all the files to our Github Repository
- Upload package and oversize file in AWS Lambda
- Lambda Configuration and Lambda trigger (Cloud Watch Event)



# Lessons Learned

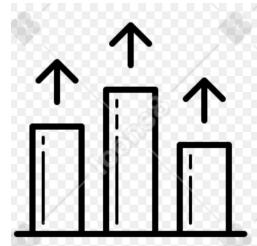


- API is very useful and powerful even with its limitations
- Importance of the ETL process
- Versatility of AWS and its multiple options
- That restaurants do not need to be clean in Manhattan in order to be successful



# Improvements

- Building Layer using cloud9 and EC2 to install packages and set up virtual environments that we need in AWS Lambda
- Creating ETL process in AWS glue using the transformation to process data
- Use multiple sources of online reviews like Google Reviews or Tripadvisor
- Finding more relevant data to provide more valuable insight





Thank you



**Katz**

Katz School  
of Science and Health