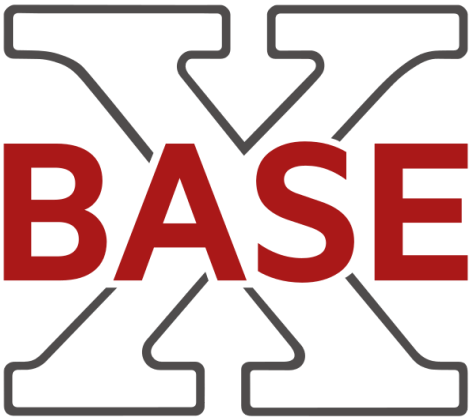
Big Data Proof of Concept: BaseX



By: TeamX
Gavin Celichowski, Jamsrandorj Dagvadorj, Shreyasi Pal, Arvind Renganathan

CSCI 5751
Professor Donald Sawyer

1. Datasets

1.1. Main Dataset: Yelp Challenge Dataset

Our data comes from the Yelp challenge dataset, which provides a chance for students to conduct research or analysis on Yelp's data and share their discoveries in exchange for the chance to earn $5,000. The Yelp dataset challenge has been going on for years, and we are using the Round 13 dataset. Round 13 runs from January 15th, 2019 through December 31st, 2019. The dataset contains six JSON files related to the different businesses, users, reviews, user checkins, tips and photo uploads. The dataset contains a subset of Yelp's businesses, reviews and user data, and it is intended for use in personal, educational, and academic purposes. The dataset is delivered as a TAR compressed archive, and consists of 6 JSON documents containing business information, reviews, user information, check-in timestamps for businesses, Yelp tips and metadata for photos. Tips and reviews are similar concepts, but they have a slightly different purpose for Yelp. Yelp tips are shorter reviews, usually just a couple sentences max in length, and reviews are typically longer assessments that are more than one paragraph in length. In the dataset you'll find information about businesses and users across 10 metropolitan areas in four countries. The Dataset can be downloaded at https://www.yelp.com/dataset/challenge. It is downloaded as a .tar archive, which must be uncompressed before using.

In total, there are:

- 6,685,900 user reviews
- Information on 192,609 businesses
- 200,000 pictures
- 10 metropolitan areas
- 1,223,094 tips by 1,637,138 users
- Over 1.2 million business attributes such as hours, parking, availability, and ambience
- Aggregated check-in timestamps accumulated over time for each of the 192,609 businesses

1.1.1. Data Dictionary

- Business.json
  Contains business data including location data, attributes, and categories.

Attribute Name	Data Type	Description
business_id	string	22 character unique string business id
name	string	the business's name
city	string	city
state	string	2 character state code, if applicable
postal code	string	the postal code
latitude	float	latitude
longitude	float	longitude
stars	float	star rating, rounded to half-stars
review_count	integer	number of reviews
is_open	integer	0 or 1 for closed or open, respectively
attributes	object	business attributes to values. note: some attribute values might be objects
categories	array of strings	business categories
hours	object	key day to value hours, hours are using a 24hr clock

- Review.json
  Contains full review text data including the user_id that wrote the review and the business_id the review is written for.

Attribute Name	Data Type	Description
review_id	string	22 character unique string review id
user_id	string	22 character unique user id, maps to the user in user.json
business_id	string	22 character business id, maps to business in business.json
stars	integer	star rating
date	string	date formatted YYYY-MM-DD
text	string	the review itself
useful	integer	number of useful votes received
funny	integer	number of funny votes received
cool	integer	number of cool votes received

- User.json
  User data including the user's friend mapping and all the metadata associated with the user.

Attribute Name	Data Type	Description
user_id	string	22 character unique string user id
name	string	the user's first name
review_count	integer	the number of reviews they've written
yelping_since	string	when the user joined Yelp, formatted like YYYY-MM-DD
friends	array of strings	an array of the user's friend as user_ids
useful	integer	number of useful votes sent by the user
funny	integer	number of funny votes sent by the user
cool	integer	number of cool votes sent by the user
fans	integer	number of fans the user has

Attribute Name	Data Type	Description
elite	array of integers	the years the user was elite
average_stars	float	average rating of all reviews
compliment_hot	integer	number of hot compliments received by the user
compliment_more	integer	number of more compliments received by the user
compliment_profile	integer	number of profile compliments received by the user
compliment_cute	integer	number of cute compliments received by the user
compliment_list	integer	number of list compliments received by the user
compliment_note	integer	number of note compliments received by the user
compliment_plain	integer	number of plain compliments received by the user
compliment_cool	integer	number of cool compliments received by the user
compliment_funny	integer	number of funny compliments received by the user
compliment_writer	integer	number of writer compliments received by the user
compliment_photos	integer	number of photo compliments received by the user

- Checkin.json
  Checkins on a business.

Attribute Name	Data Type	Description
business_id	string	22 character business id, maps to business in business.json
date	string	a comma-separated list of timestamps for each checkin. (YYYY-MM-DD HH:MM:SS)

- Tip.json
  Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.

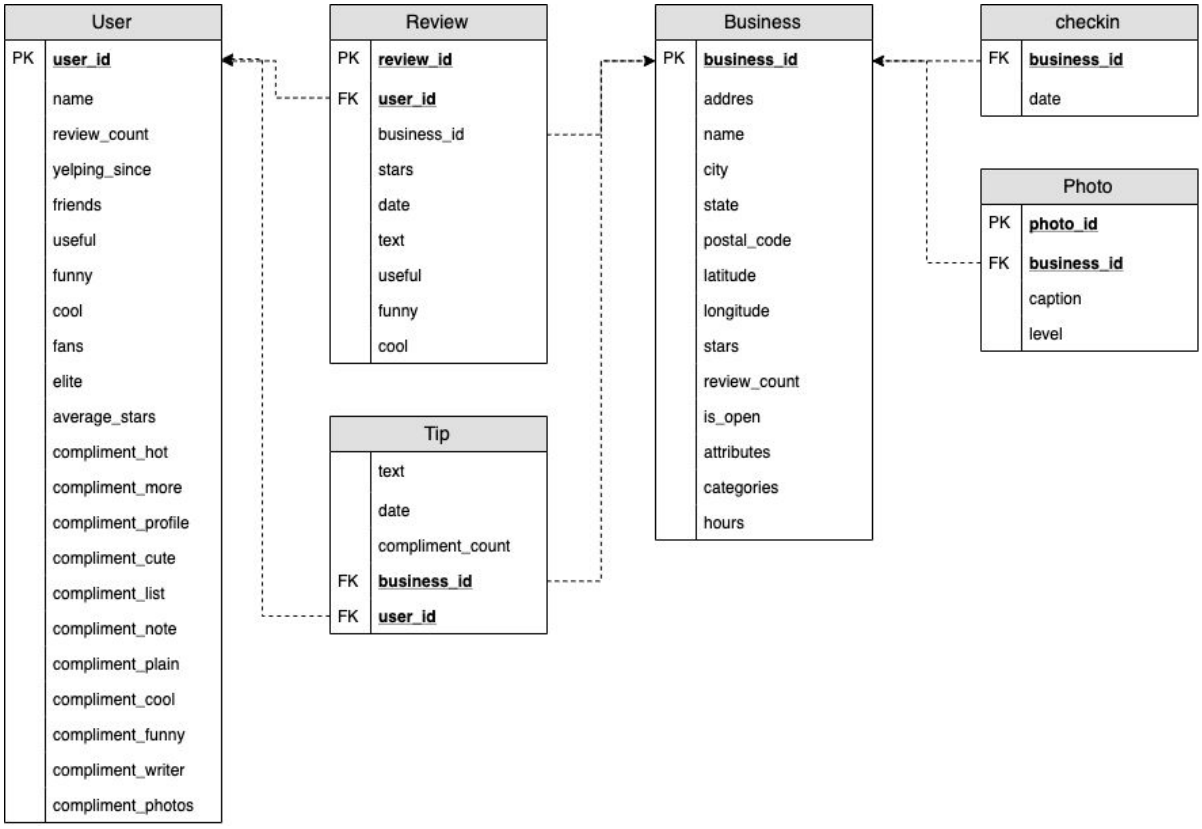
Attribute Name	Data Type	Description
text	string	text of the tip
date	string	when the tip was written, formatted like YYYY-MM-DD
compliment_count	integer	how many compliments it has
business_id	string	22 character business id, maps to business in business.json
user_id	string	22 character unique user id, maps to the user in user.json

- Photo.json
  Contains photo data including the caption and classification (one of "food", "drink", "menu", "inside" or "outside").

(We identified this is not required to answer any of the business questions. Photo.json was dropped in this step in identifying data stage)

Attribute Name	Data Type	Description
photo_id	string	22 character unique photo id
business_id	string	22 character business id, maps to business in business.json
caption	string	the photo caption, if any
label	string	the category the photo belongs to, if any

1.1.2. Dataset Breakdown Diagram



1.2. Data Modeling

On a large scale, we maintained one data model, which included all of the cleaned XML-formatted data. The data model can be seen in Section 1.1.2, but we also used multiple smaller data models. Many of our business questions required joining of data from multiple documents, and through trial and error we realized the best way to join data together was to create smaller data models using XQuery, and then join the data at the application layer in Python. Additionally, we created a new data model featuring new keys for Challenging Business Question 2. The new key was the category and the value was the array of locations of businesses that came under it. Along with the new keys, the categories were extracted and further processed. The business questions in Sections 5.2.3 and 5.3.4 also use this method of data modeling.

1.3. Supplementary Dataset

In addition to dataset, we leveraged another dataset to enhance our analysis. The dataset we used was a set of restaurant health inspection data that we use to see if user ratings of

restaurants correlate at all to their health inspection grades. The health inspection dataset can be downloaded from here.

2. NoSQL Storage Technology

The storage technology we used to persist and analyze the Yelp Challenge Dataset is BaseX. BaseX is an XML document database. BaseX has the following features:

- XQuery/XPath query language processor
- Default indexing of all attributes and their paths
    - Customizable indexing
- Graphical and Command Line interfaces
- REST API for querying

3. Project Life Cycle Overview

3.1. Business Questions: (Section 5)
3.2. Identify Data (Section 1.1)
3.3. Acquire and Filter Data (Section 4)
3.4. Extract data (Section 4)
3.5. Validate and cleanse (Section 4)
3.6. Aggregate and structure (Section 4)
3.7. Analyze (Section 5)
3.8. Visualize (Section 5)
3.9. Utilize Results

4. Data Manipulations: Clean/Filter/Transform

BaseX has many powerful built in tools to easily ingest documents and convert them to XML, but the following issues required data engineering efforts:

- XQuery's concept of NULL (empty sequence) was incompatible with our data set
- BaseX requires perfectly formed documents in order to ingest them, and our dataset's JSON documents had some JSON objects that were not correctly formed
- While using its built in JSON to XML conversion algorithms, BaseX attempts to load the entire document into memory
    - Due to the size of some of the documents, this was impossible

With the constraints of our technology and dataset in mind, we took the following steps to clean and filter our data, and the code associated with each step can be found in the appendix:

- Counted the number of NULL values in each column in each document (here)
- Cleared out the rows containing NULL data from each document (here)
- Fixed the JSON objects in the documents that were missing curly braces (here)

- Converted the documents from JSON to XML outside of BaseX (here)
- Adding provenance along the steps in the process

After these steps were completed, we were able to ingest the Yelp Dataset into BaseX.

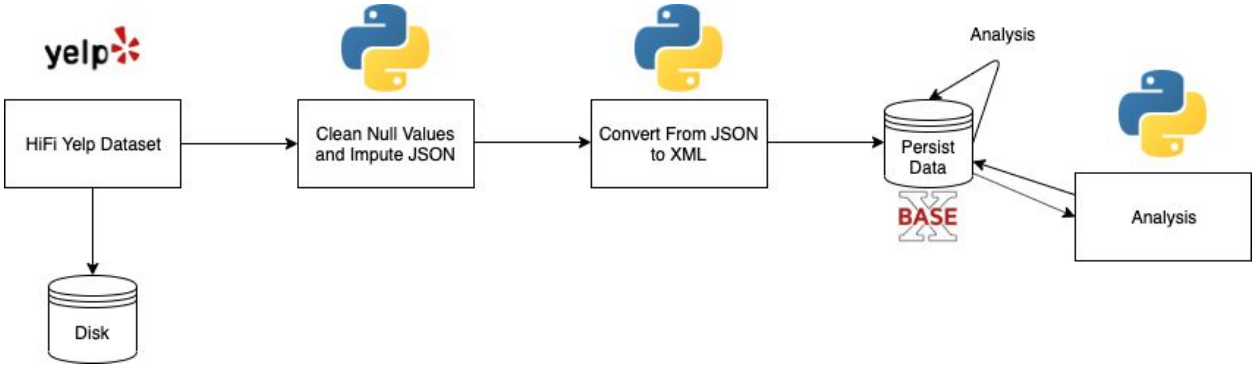


Figure 1: Data Flow Diagram

5. Results of Analytical Questions

The business questions help to get insights about different business and customer patterns. We also used visualization and statistical techniques to further analyze the data.
The code associated with Business Questions can be found on GitHub. The code for each individual question is linked in Appendix A.

5.1. Easy Questions

5.1.1. What is the range/min/max/average stars for the businesses? (See Table 1)

5.1.2. What is the range/min/max/average stars for the reviews? (See Table 1)

5.1.3. What is the range/min/max/average review counts for the businesses? (See Table 1)

Name	Average	Min	Max	Range
Stars (Business)	3.7126435003249063	1	5	4
Stars (Review)	3.7160564576687163	1	5	4
Review Count (Business)	44.50837020763066	3	8348	8345

Table 1: Results to Questions 5.1.1-5.1.3

5.1.4. How many of each compliment type does each user have?

Compliment (Per User)	Avg	Min	Max	Range
Total	22.473000071851175	0	277077	277077
Cool	4.331028089679694	0	32266	32266
Cute	0.25565995832598487	0	13654	13654
Funny	.3310280896796944	0	32266	32266
Hot	3.1396291796489715	0	34167	34167
List	0.1157003694921823	0	12669	12669
More	0.4410020924100601	0	13500	13500
Note	2.0362639049466664	0	57833	57833
Photos	1.6020654112206305	0	82602	82602
Plain	4.3030059070848	0	52103	52103
Profile	0.29869643506452503	0	14173	14173
Writer	1.6189206342985418	0	15442	15442

Table 2: Results to Question 5.1.4

5.1.5. When do people tend to check-in?

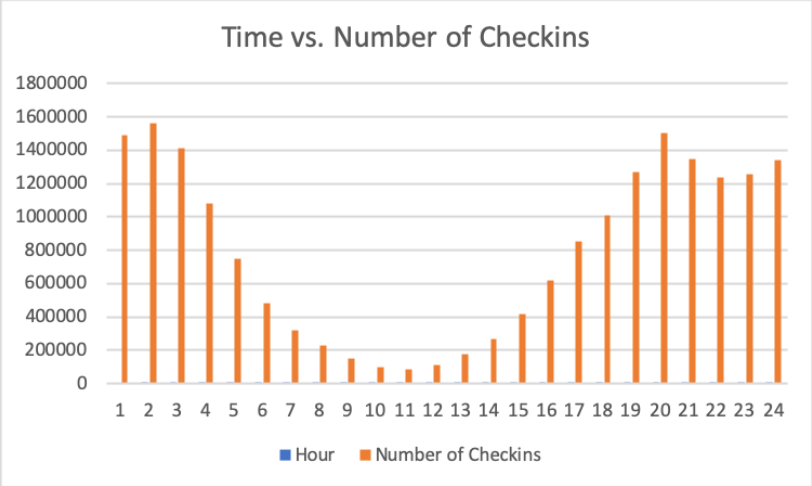


Figure 1: Chart Corresponding to Question 5.1.5

5.1.6. How many NULL values do columns have? Does it make more sense to try to impute the data for those columns or delete the columns?

- Business.json was the only document with NULL values, and these were the results
  - Categories is missing values for only a small percentage of businesses, so we decided to leave it alone
  - Attributes are difficult to impute (even for chains, as their offerings can vary restaurant to restaurant)
  - We attempted to impute Address and Hours using the Google Maps API, but only opening hours can be imputed via Google Maps
    - Additionally, we were only able to get results for ~30% of the missing addresses, and there is no way to verify correctness. Based on these factors, we decided to exclude the imputed data; we do not want to use possibly incorrect data in our analysis
  - We thought of imputing the Hours by aggregating the time of creation of the photos of each business, but we found that creation and last modification timestamp were modified when Yelp created the dataset and when we extracted the tar file.

Attribute	Number of Null Values
Address	7682
Attributes	28836
Business_id	0
Categories	482
City	1
Hours	44830
Is_open	0
Latitude	0
Longitude	0
Name	0
Postal_code	659
Review_count	0
stars	0
state	0

Table 3: Results to Question 5.1.6

5.2. Moderately Difficult Questions

5.2.1. How does a restaurant in one metropolitan area compare to the same business in another area (assuming they are a chain)?

Business name	Number of restaurant	Average rating	State with max rating		State with min rating	
			state	rating	state	rating
Starbucks	885	3.2542	QC	3.56756	IL	2.9166
McDonald's	717	2.036959	NY	2.5	IL	1.7
Subway	331	2.7462	PA	3	IL	1.5
Walgreens	294	2.7551	PA	3.62	IL	2.5
Taco Bell	263	2.4695	NV	2.7058	WI	2.03571
The UPS Store	240	3.218	NC	3.2941	ON	2.75
CVS Pharmacy	239	2.7217	IL	3.25	OH	2.5645
Circle K	207	2.5797	NV	3.0555	AZ	2.4818
Domino's Pizza	191	2.6753	ON	3.0952	PA	2.0833
KFC	182	2.0109	PA	2.944	WI	1.571

Table 4: Results to Business Question 5.2.1

5.2.2. Which users are considered more reliable reviewers (i.e. Which users have more useful/fun/cool reviews (top 5%))?

Review Rating Category	Top 5% users in category
“Funny” People	['Roland', 'A', 'Jay', 'G.', 'Eric', u'A\xefcha', 'Jeremy', 'Myisha', 'Grace', 'Kelly', 'Rob', 'Jacqueline', 'William', 'Mike', 'Tim', 'Jennifer', 'John', 'Dylan', 'Emma', 'Brianna', 'Casandra', 'Christopher', 'Jackie', 'David', 'Becky', 'Rob', 'Jerry', 'Molinna', 'Adrian', 'Roland', 'A', 'Jay', 'G.', 'Eric', 'Nicole', 'Chris', 'Jason', 'Oh', 'Jake', 'Geraldine', 'Down Town', 'Betty', 'skip', 'Shelby', 'Jordan', 'Yesenia', 'Ross', 'Robin', 'Barbara', 'Cruz', 'Karen', 'Sanjeev', 'Lorena', 'Jessica', 'Raul', 'Stephanie', 'Ally', 'Katelyn']
“Useful” People	['Roland', 'Shanna', 'Marian', 'Denise', 'Elle', 'A And', 'Grace', 'Kelly', 'Aide', 'Jacqueline', 'William', 'Wendy', 'Dave', 'Pravee', 'V', 'Brianna', 'Meelad', 'Maria', 'oggie', 'Kevin', 'Veganos', 'Jerry', 'Ron', 'Roland', 'Shanna', 'Marian', 'Denise', 'Seth', 'Oh', 'Down Town', 'Joe', 'Betty', 'S', 'jorge', 'Yesenia', 'Rosanna', 'Terrence', 'Ross', 'Tammy', 'Moe', 'Kristin', 'Cruz', 'Michelle', 'Nicole', 'Enrique', 'Sanjeev', 'Lorena', 'Jessica', 'Andie', 'Katelyn']
“Cool” People	['Michelle', 'G.', 'Miriam', 'A And', 'Grace', 'Kelly', 'Hon', 'Ashley', 'William', 'Bruce', 'Kelsey', 'Brian', 'Brianna', 'Casandra', 'Christopher', 'Jackie', 'Becky', 'Jerry', 'Molinna', 'Adrian', 'Michelle', 'G.', 'Miriam', 'Usha', 'Geraldine', 'Down Town', 'Betty', 'Yesenia', 'Terrence', 'Ross', 'Robin', 'Barbara', 'Francesca', 'Cruz', 'Karen', 'Sanjeev', 'Jessica', 'Kristi', 'Katelyn']

Table 5: Results to Business Question 5.2.2

5.2.3. Which businesses have higher/lower rating in terms of business category?

Business Name	City	State	Review Count
Computer Doctor BG	Las Vegas	Nevada	211
One Shot Installation	Peoria	Arizona	114
Rapid iPhone Repair	Gilbert	Arizona	112
Tech Mail	Las Vegas	Nevada	141
Fixitup iPhone and iPad Repair	Las Vegas	Nevada	197
GadgetMates	Las Vegas	Nevada	137
Computer Repair Las Vegas	Las Vegas	Nevada	118

Table 6: Results to Business Question 5.2.2, IT Service Providers with 5 Star Ratings and Top 1% Review Counts

5.2.4. Are there some attributes that affect on business rating?

ex. Do businesses with WiFi have higher rating than those with no Wifi?

Attribute	All		True/Yes		False/No	
	avg(rating)	N	avg(rating)	N	avg(rating)	N
WiFi	3.56615	41773	3.56483	30037	3.5695296	11736
GoodForKids	3.6268790	51490	3.624580	41467	3.636386	10023
Restaurants Delivery	3.539630	42152	3.553436	11406	3.5345085	30746
Caters	3.5838481	35129	3.6635967	18567	3.494445	16562
Restaurants Reservations	3.5272	41765	3.59387	17123	3.480926	24642
BikeParking	3.707706	72304	3.7465	57434	3.55766	14870

Table 7.a: Results to Business Question 5.2.5

Attribute	All		True/Yes		False/No	
	avg(rating)	N	avg(rating)	N	avg(rating)	N
OutdoorSeating	3.5453	43335	3.59243	18440	3.510443	24895
RestaurantsTableService	3.6725	15720	3.6937	9988	3.63555	5732
RestaurantsGoodForGroups	3.51933	42684	3.51933	37593	3.50756	5091
BusinessAcceptsCreditCards	3.76278	93492	3.7553	88843	3.904818	4649
RestaurantsTakeOut	3.55772	49327	3.54806	44889	3.65547	4438
NoiseLevel	3.5378384	35863	-	-	-	-

Table 7.b: Results to Business Question 5.2.5

NoiseLevel	N	Average rating
quiet	7474	3.65192
average	23834	3.55504
loud	3199	3.321350
Very loud	1302	3.096006

Table 8: Results to Business Question 5.2.5

5.3. Challenging Questions

5.3.1. Why do the different metropolitan areas have different tendencies in which businesses they like/dislike?

We found some trends between states and businesses, such as Pennsylvania having a higher rating in fast food chain restaurants such as KFC, Subway, etc. while Illinois tends to have lower ratings for those restaurants. Though we could not find the cause for these trends, we thought they would be a point of interest for future analysis.

5.3.2. Plot the business locations according to their business categories on a map

For this question, we focused our analysis on the businesses that are open in USA.

- The information related to the businesses that are open were extracted using XQuery
- Each business is associated with multiple categories. The categories were preprocessed (refer to section preprocessing)
- Apriori algorithm and association rules were applied to the categories to understand the patterns. We created the itemsets that had support greater than 0.01 and then we created the association rules where the confidence was 1 which gave us the categories which are complete subsets.
- The consequents whose length = 1 were chosen as the main categories. The main categories are: ['Beauty & Spas', 'Local Services', 'Restaurants', 'Bars', 'Event Planning & Services', 'Automotive', 'Nightlife', 'Shopping', 'Home Services', 'Health & Medical', 'Food', 'Active Life']
- 3 of them were chosen for visualization: ['Food', 'Automotive', 'Home Services']

Legend:

- **Red:** Restaurant
- **Green:** Automotive
- **Blue:** Home Services

(gmplot does not have a method for adding legends yet)

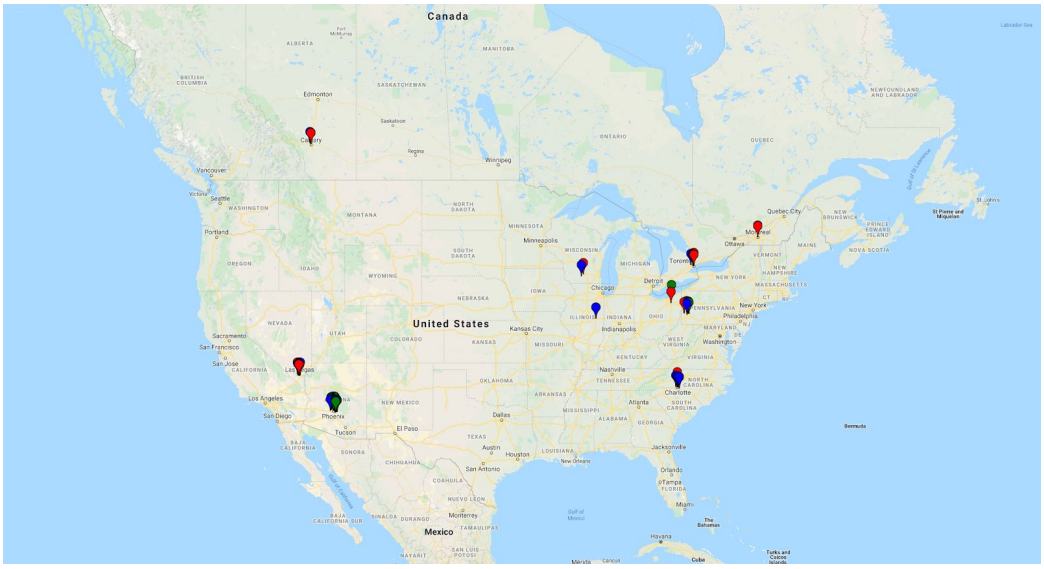


Figure 2.a: Plot Corresponding to Question 5.3.3

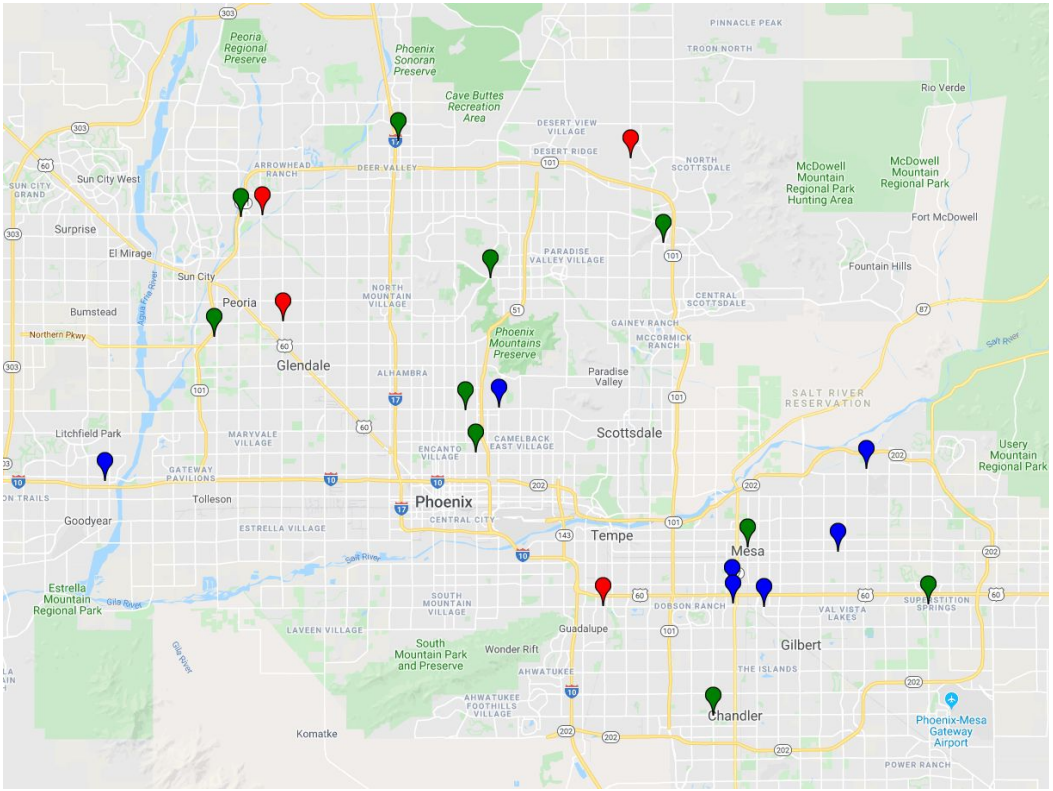


Figure 2.b: Plot Corresponding to Question 5.3.3

5.3.3. Can you tell whether a review is positive or negative based on review text?

Based on these results, we can determine that specific words have definite effects on the rating for the business.

words	Average review rating with a word
bad	2.8466
good	3.72270
great	4.2218
delicious	4.39
disappointed	3.07
horrible	1.651

Table 9 : Results to Business Question 5.3.3

*average review rating = 3.71.

5.3.4. How does the Environmental Health Safety Inspection rating for Las Vegas businesses correlate to their Yelp ratings?

Based on the data in the two sets, the Environmental Health Safety Inspection Rating of a Restaurant is not very correlated to the Yelp Rating

Grade	Average Stars	Number of Restaurants with that Grade
A	3.4774266365688487	443
B	2.875	8
C	3.6666666666666665	3

Table 10: Results to Business Question 5.3.4

6. Lessons Learned

- BaseX is a powerful tool, but has its limitations
    - Data volume created memory and query speed issues
    - Index heavy, writing documents takes a LONG time
    - Built in processing tools proved insufficient for big data
- XQuery
    - XPath very useful for subsetting large documents
    - Aggregate queries for attributes work very well due to indexes
    - Joining data is inefficient and often exceeds maximum heap allocation
- Processing Big Data
    - Chunking is imperative for large files
    - Pandas is very useful for cleaning/filtering data
    - Application layer analysis

7. Challenges Faced and Workarounds

- Challenge: BaseX could not parse JSON objects that were missing curly braces or were otherwise malformed
    - Workaround: Write a Python script to replace missing curly braces or delete rows that were misformed (cleaning the data)
- Challenge: BaseX's built in JSON to XML converter would use all of main memory and fail
    - Workaround: Convert from JSON to XML in Python
- Challenge: Python would also fail converting documents from JSON to XML due to memory constraints
    - Workaround: Process the data in chunks
        i. We also used this strategy for challenge 01
- Challenge: 38% of the Address attributes were NULL in Business.json
    - Workaround: We wrote a Python script to attempt to impute the address of the restaurant based on its latitude and longitude using the Google Maps API, but only ~30% of the Google Maps API requests returned a result, and there was no way to be certain we had imputed the correct address, so we ultimately ended up not using any of the imputed address
    - Additional Challenge: Google Maps API is limited to 100,000 requests per day on the free subscription, and therefore we had limited attempts to test our algorithm at full scale
- Challenge: Our business questions required many different tables to be related, and denormalization would not solve all of our problems
    - Note: XQuery could theoretically support a join (nested For, Let, Where, Order By, Return (FLWOR) statements), these operations always caused BaseX to run out of heap space due to data volume and XQuery's underlying algorithms

    - Workaround: Use XQuery to select subsets of our datasets, save the subsets to files, and then simulate joins on the smaller datasets in Python, leveraging its wider range of data structures and algorithmic capabilities

8. What Worked/Did Not Work

This project led us to do a lot of trial and error with different technologies and strategies, and most of our problems led to us deciding between implementing something in the database layer within BaseX or using Python in the application layer. While we got to leverage some of BaseX's powerful built in tools (e.g. automatic indexing of all attributes), we also relied heavily on the application layer for our data cleaning and filtering. Additionally, the application layer was very helpful in our analytical pursuits. BaseX queries often executed very slowly due to the volume of data, and we found that Python was typically faster for churning through data and joining multiple datasets together.

- What Worked:
    - Pandas to summarize datasets
    - Python and chunking to process datasets
    - XQuery for simple analytical and aggregate queries
    - XQuery for creating smaller data models with subsets of documents, and then processing them in Python
- What Did Not Work:
    - BaseX's built in JSON-to-XML conversion
    - XQuery for complex queries or queries spanning large datasets

9. Appendix A: Queries/Programs Used to Answer Business Questions

- All questions are links to their solution in GitHub

9.1. Easy Questions

9.1.1. What is the range/min/max/average stars for the businesses?

9.1.2. What is the range/min/max/average stars for the reviews?

9.1.3. What is the range/min/max/average review counts for the businesses?

9.1.4. How many of each compliment type does each user have?

9.1.5. When do people tend to check-in?

9.1.6. How many NULL values do columns have? Does it make more sense to try to impute the data for those columns or delete the columns?

9.2. Moderately Difficult Questions

9.2.1. How does a restaurant in one metropolitan area compare to the same business in another area (assuming they are a chain)?

9.2.2. Which users are considered more reliable reviewers (i.e. Which users have more useful/fun/cool reviews (top 5%))?

9.2.3. Which businesses have higher/lower rating in terms of business category?

9.2.4. Are there some attributes that affect on business rating?

9.3. Challenging Questions

9.3.1. Why do the different metropolitan areas have different tendencies in which businesses they like/dislike?

9.3.2. Plot the business locations according to their business categories on a map

9.3.3. Can you tell whether a review is positive or negative based on review text?