Yelp Dataset Challenge

Priyanka Sudhindra - 422284041

Abstract— ***This paper provides an analysis of the Yelp dataset which includes information about local businesses in 10 metropolitan areas across 2 countries. The Yelp dataset is publicly available as a part of the Yelp Dataset Challenge.***

I. Introduction

By exploring the Yelp dataset, we are trying to answer a few questions like: (1) What are the top categories of restaurants? (2) Is there a correlation between the price range a restaurant falls under and its average rating? (3) Is there a correlation between price and ratings? (4) Where are the maximum number of 5-star restaurants located? (5) What are the top categories that most of the 5 star rated restaurants fall under? (6) Which city is the kid-friendly paradise? (7) Can we show an interactive map of restaurants in the city with an indication of their ratings? (8) What is the comparison between the maximum number of reviews for a restaurant and the number of high ratings? (9) Which neighborhood houses the maximum number of highly rated restaurants? (10) Can we create a word cloud of the most frequently occurring phrases for the low rated restaurants?

II. Prior Work

This project is an add on to already existing sources of Yelp dataset analysis provided by participants of the Yelp Dataset Challenge over the years. This project does not involve prediction of any kind, bus, uses the dataset provided by Yelp to discover patterns in the culinary world across North America.

III. Data Description

The dataset used in this project is provided by Yelp as a part of the Yelp Dataset Challenge 2018. The dataset includes information about the business, checkin, reviews, user and tip.

Since the dataset provided is huge and contains millions of records, for most of the analysis, we have restricted ourselves to using data pertaining to the city of Las Vegas.

However, the same procedure can be followed to analyze restaurants of other cities of the world.

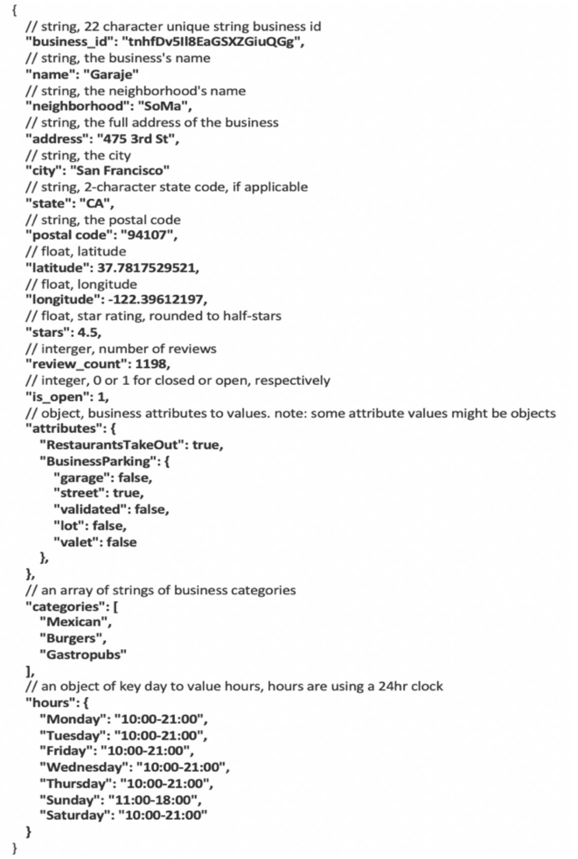
The dataset contains:

- 5996996 reviews
- 188593 business
- 157075 checkin
- 1140055 tips
- 1326102 users
- 10 metropolitan areas

Each file is composed of a single object type, one JSON-object per-line.

1. *business.json*

Contains business data including location data, attributes and categories.

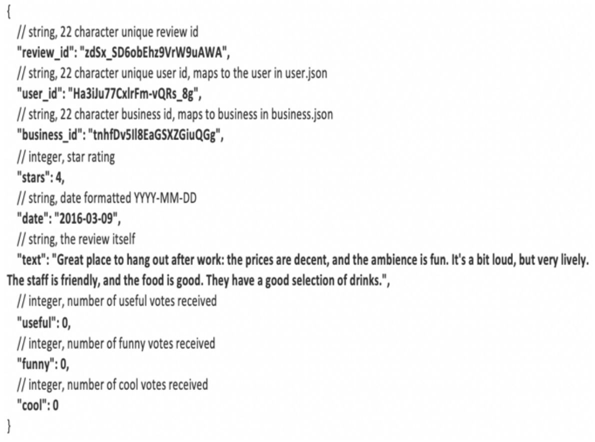


```
{
  // string, 22 character unique string business id
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg",
  // string, the business's name
  "name": "Garaje"
  // string, the neighborhood's name
  "neighborhood": "SoMa",
  // string, the full address of the business
  "address": "475 3rd St",
  // string, the city
  "city": "San Francisco"
  // string, 2-character state code, if applicable
  "state": "CA",
  // string, the postal code
  "postal code": "94107",
  // float, latitude
  "latitude": 37.7817529521,
  // float, longitude
  "longitude": -122.39612197,
  // float, star rating, rounded to half-stars
  "stars": 4.5,
  // interger, number of reviews
  "review_count": 1198,
  // integer, 0 or 1 for closed or open, respectively
  "is_open": 1,
  // object, business attributes to values. note: some attribute values might be objects
  "attributes": {
    "RestaurantsTakeOut": true,
    "BusinessParking": {
      "garage": false,
      "street": true,
      "validated": false,
      "lot": false,
      "valet": false
    },
  },
  // an array of strings of business categories
  "categories": [
    "Mexican",
    "Burgers",
    "Gastropubs"
  ],
  // an object of key day to value hours, hours are using a 24hr clock
  "hours": {
    "Monday": "10:00-21:00",
    "Tuesday": "10:00-21:00",
    "Friday": "10:00-21:00",
    "Wednesday": "10:00-21:00",
    "Thursday": "10:00-21:00",
    "Sunday": "11:00-18:00",
    "Saturday": "10:00-21:00"
  }
}
```

Figure 1. A typical business.json file

2. *review.json*

Contains text reviews for business bearing a *business_id* written by user with a *user_id*.

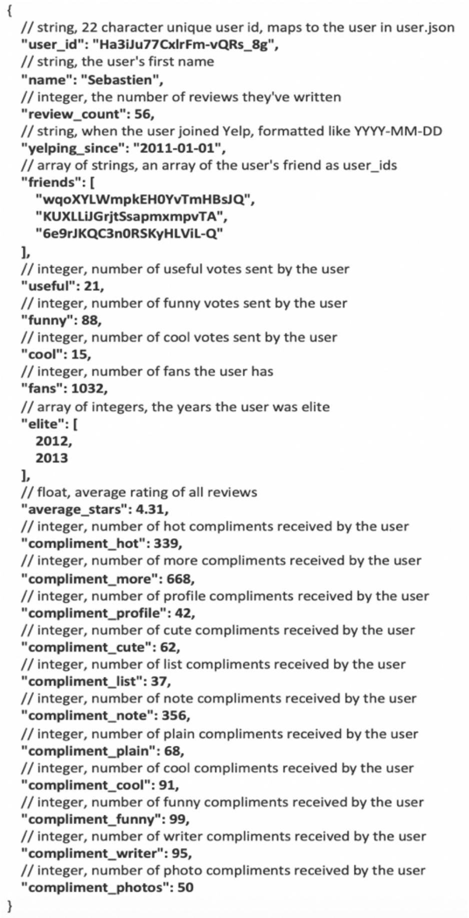


```
{
  // string, 22 character unique review id
  "review_id": "zdSx_SD6obEhz9VrW9uAWA",
  // string, 22 character unique user id, maps to the user in user.json
  "user_id": "Ha3iJu77CxlrFm-vQRs_8g",
  // string, 22 character business id, maps to business in business.json
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg",
  // integer, star rating
  "stars": 4,
  // string, date formatted YYYY-MM-DD
  "date": "2016-03-09",
  // string, the review itself
  "text": "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively.
The staff is friendly, and the food is good. They have a good selection of drinks.",
  // integer, number of useful votes received
  "useful": 0,
  // integer, number of funny votes received
  "funny": 0,
  // integer, number of cool votes received
  "cool": 0
}
```

Figure 2. A typical review.json file

3. *user.json*

Contains user data including the user’s friend mapping and all the metadata associated with the user.

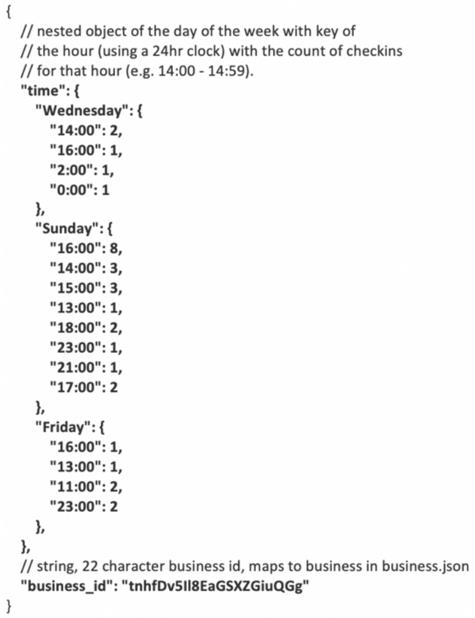


```
{
  // string, 22 character unique user id, maps to the user in user.json
  "user_id": "Ha3iJu77CxlrFm-vQRs_8g",
  // string, the user's first name
  "name": "Sebastien",
  // integer, the number of reviews they've written
  "review_count": 56,
  // string, when the user joined Yelp, formatted like YYYY-MM-DD
  "yelping_since": "2011-01-01",
  // array of strings, an array of the user's friend as user_ids
  "friends": [
    "wqoXYLWmpkEH0YvTmHBsJQ",
    "KUXLLiJGrjtSsapmxmpvTA",
    "6e9rJKQC3n0RSKyHLViL-Q"
  ],
  // integer, number of useful votes sent by the user
  "useful": 21,
  // integer, number of funny votes sent by the user
  "funny": 88,
  // integer, number of cool votes sent by the user
  "cool": 15,
  // integer, number of fans the user has
  "fans": 1032,
  // array of integers, the years the user was elite
  "elite": [
    2012,
    2013
  ],
  // float, average rating of all reviews
  "average_stars": 4.31,
  // integer, number of hot compliments received by the user
  "compliment_hot": 339,
  // integer, number of more compliments received by the user
  "compliment_more": 668,
  // integer, number of profile compliments received by the user
  "compliment_profile": 42,
  // integer, number of cute compliments received by the user
  "compliment_cute": 62,
  // integer, number of list compliments received by the user
  "compliment_list": 37,
  // integer, number of note compliments received by the user
  "compliment_note": 356,
  // integer, number of plain compliments received by the user
  "compliment_plain": 68,
  // integer, number of cool compliments received by the user
  "compliment_cool": 91,
  // integer, number of funny compliments received by the user
  "compliment_funny": 99,
  // integer, number of writer compliments received by the user
  "compliment_writer": 95,
  // integer, number of photo compliments received by the user
  "compliment_photos": 50
}
```

Figure 3. A typical user.json file

4. *checkin.json*

Contains the checkins on a business.

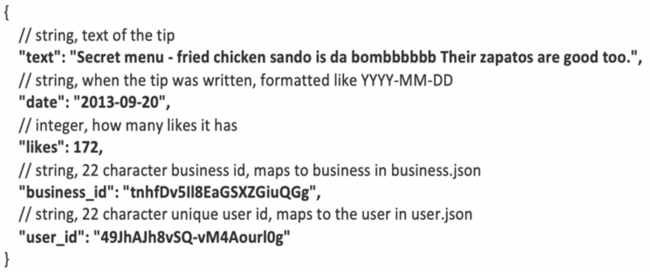


```
{
  // nested object of the day of the week with key of
  // the hour (using a 24hr clock) with the count of checkins
  // for that hour (e.g. 14:00 - 14:59).
  "time": {
    "Wednesday": {
      "14:00": 2,
      "16:00": 1,
      "2:00": 1,
      "0:00": 1
    },
    "Sunday": {
      "16:00": 8,
      "14:00": 3,
      "15:00": 3,
      "13:00": 1,
      "18:00": 2,
      "23:00": 1,
      "21:00": 1,
      "17:00": 2
    },
    "Friday": {
      "16:00": 1,
      "13:00": 1,
      "11:00": 2,
      "23:00": 2
    },
  },
  // string, 22 character business id, maps to business in business.json
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg"
}
```

Figure 4. A typical checkin.json file

5. *tip.json*

Contains quick suggestions written by users. They are shorter than reviews.



```
{
  // string, text of the tip
  "text": "Secret menu - fried chicken sando is da bombbbbbb Their zapatos are good too.",
  // string, when the tip was written, formatted like YYYY-MM-DD
  "date": "2013-09-20",
  // integer, how many likes it has
  "likes": 172,
  // string, 22 character business id, maps to business in business.json
  "business_id": "tnhfDv5Il8EaGSXZGiuQGg",
  // string, 22 character unique user id, maps to the user in user.json
  "user_id": "49JhAJh8vSQ-vM4Aourl0g"
}
```

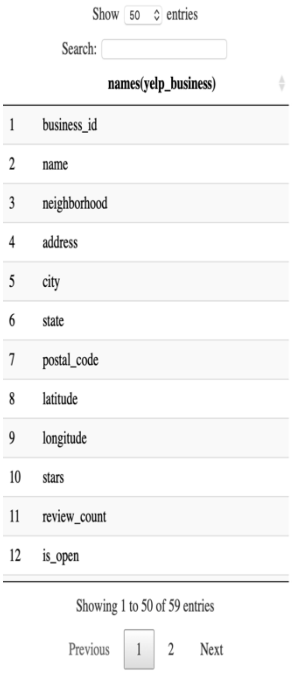
Figure 5. A typical tip.json file

IV. Methodologies Employed

We have made use to the Amazon Web Services’ Simple Storage Service to store the datasets. We started off by cleaning the data post which we performed wrangling using dplyr and tidyr functions to convert the data into a tidier format before performing analysis. Apart from wrangling, we also performed slicing and dicing.

Dataset Locations:

- https://s3.amazonaws.com/priyanka.yelp/yelp_academic_dataset_business.json
- https://s3.amazonaws.com/priyanka.yelp/yelp_academic_dataset_checkin.json
- https://s3.amazonaws.com/priyanka.yelp/yelp_academic_dataset_review.json
- https://s3.amazonaws.com/priyanka.yelp/yelp_academic_dataset_tip.json
- https://s3.amazonaws.com/priyanka.yelp/yelp_academic_dataset_user.json



Show 50 entries

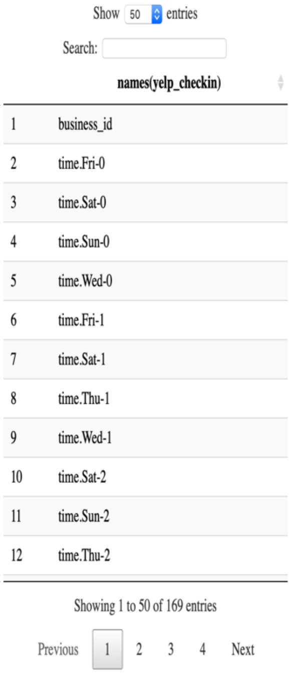
Search:

	names(yelp_business)
1	business_id
2	name
3	neighborhood
4	address
5	city
6	state
7	postal_code
8	latitude
9	longitude
10	stars
11	review_count
12	is_open

Showing 1 to 50 of 59 entries

Previous 1 2 Next

Fig 6. business.json data table



Show 50 entries

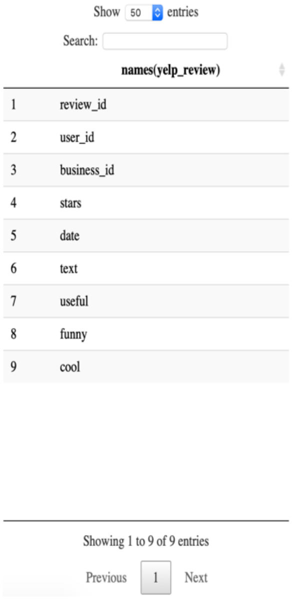
Search:

	names(yelp_checkin)
1	business_id
2	time.Fri-0
3	time.Sat-0
4	time.Sun-0
5	time.Wed-0
6	time.Fri-1
7	time.Sat-1
8	time.Thu-1
9	time.Wed-1
10	time.Sat-2
11	time.Sun-2
12	time.Thu-2

Showing 1 to 50 of 169 entries

Previous 1 2 3 4 Next

Fig 7. checkin.json data table



Show 50 entries

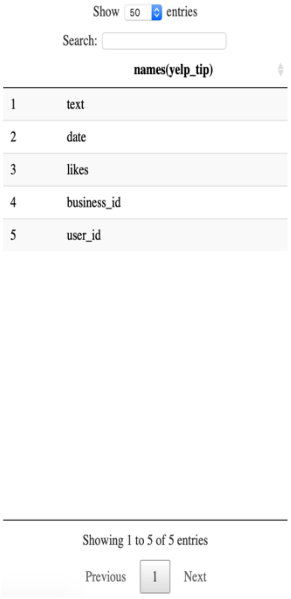
Search:

	names(yelp_review)
1	review_id
2	user_id
3	business_id
4	stars
5	date
6	text
7	useful
8	funny
9	cool

Showing 1 to 9 of 9 entries

Previous 1 Next

Fig 8. review.json data table



Show 50 entries

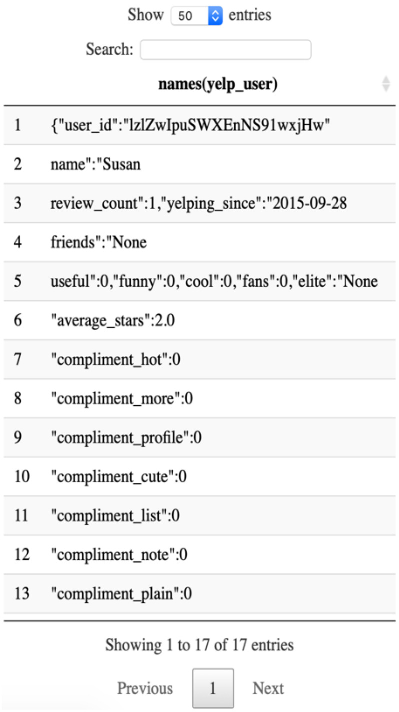
Search:

	names(yelp_tip)
1	text
2	date
3	likes
4	business_id
5	user_id

Showing 1 to 5 of 5 entries

Previous 1 Next

Fig 9. tip.json data table



Show 50 entries

Search:

	names(yelp_user)
1	{"user_id":"lzlZwIpuSWXEnNS91wxjHw"
2	name":"Susan
3	review_count":1,"yelping_since":"2015-09-28
4	friends":"None
5	useful":0,"funny":0,"cool":0,"fans":0,"elite":"None
6	"average_stars":2.0
7	"compliment_hot":0
8	"compliment_more":0
9	"compliment_profile":0
10	"compliment_cute":0
11	"compliment_list":0
12	"compliment_note":0
13	"compliment_plain":0

Showing 1 to 17 of 17 entries

Previous 1 Next

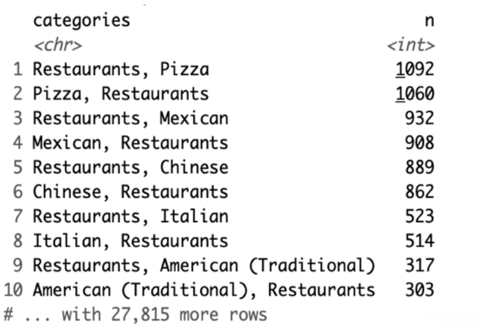
Fig 8. user.json data table

V. Experimental Setup

In this paper, we try to answer the questions listed in the Introduction section.

We have used the ggplot package to plot graphs such as boxplot, stacked bar chart, and used the leaflet package to obtain a graphical representation of the cities (map).

1. *What are the top categories of restaurants?*



```
   categories                                   n
   <chr>                                    <int>
 1 Restaurants, Pizza                        1092
 2 Pizza, Restaurants                        1060
 3 Restaurants, Mexican                       932
 4 Mexican, Restaurants                       908
 5 Restaurants, Chinese                       889
 6 Chinese, Restaurants                       862
 7 Restaurants, Italian                       523
 8 Italian, Restaurants                       514
 9 Restaurants, American (Traditional)        317
10 American (Traditional), Restaurants        303
# ... with 27,815 more rows
```

2. *Is there a correlation between the price range a restaurant falls under and its average rating?*

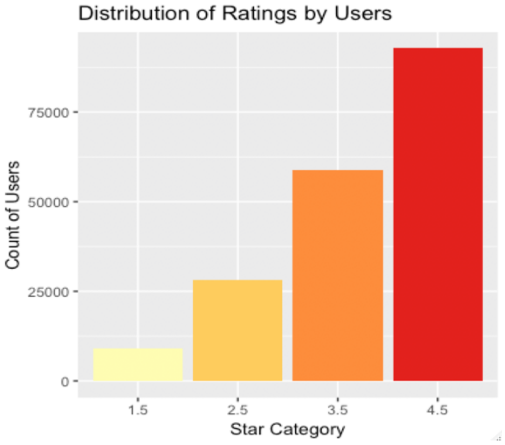


Fig 9. Distribution of Ratings by users

From the above chart we can conclude that users are more likely to review positive experiences as compared to negative ones.

3. *Is there a correlation between price and ratings?*

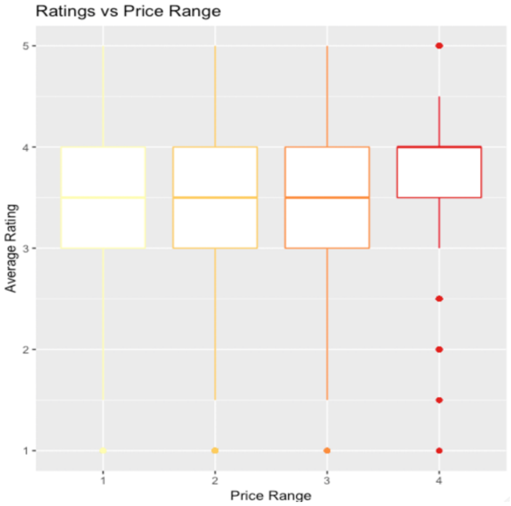
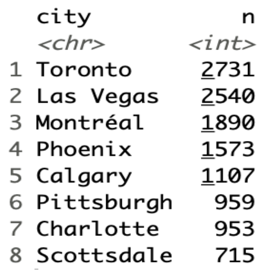


Fig10. Correlation between price and ratings

We can see that the restaurants in the price bracket of 4, which is the highest, have a higher average rating than other restaurants.

4. *Where are the maximum number of 5-star restaurants located?*



```
    city            n
    <chr>       <int>
  1 Toronto      2731
  2 Las Vegas    2540
  3 Montréal     1890
  4 Phoenix      1573
  5 Calgary      1107
  6 Pittsburgh    959
  7 Charlotte     953
  8 Scottsdale    715
```

Fig11. Top 8 5-star rated restaurants

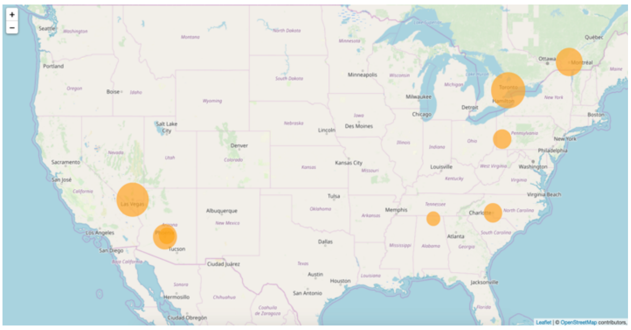
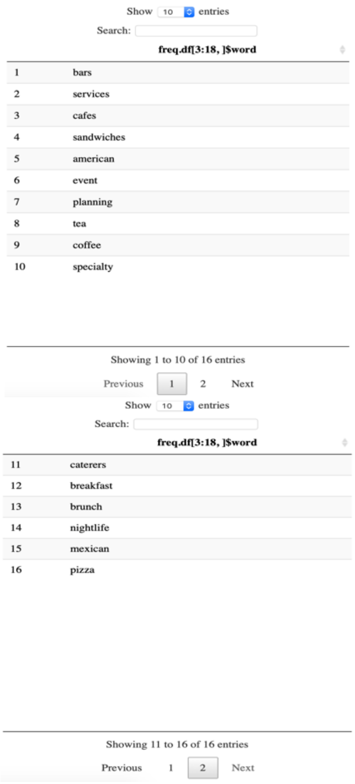


Fig12. A map showing the location of the top 8 5-star rated restaurants

5. *What are the top categories that most of the 5 star rated restaurants fall under?*



Show 10 entries

Search:

	freq.df[3:18,]$word
1	bars
2	services
3	cafes
4	sandwiches
5	american
6	event
7	planning
8	tea
9	coffee
10	specialty

Showing 1 to 10 of 16 entries

Previous 1 2 Next

Show 10 entries

Search:

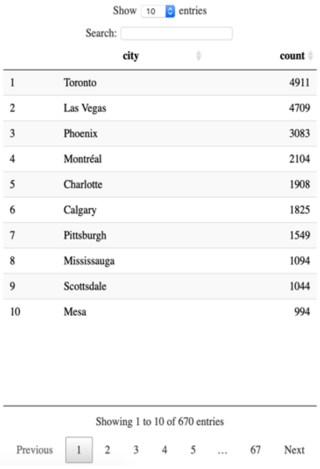
	freq.df[3:18,]$word
11	caterers
12	breakfast
13	brunch
14	nightlife
15	mexican
16	pizza

Showing 11 to 16 of 16 entries

Previous 1 2 Next

Fig 13. Top categories of 5-star rated restaurants

6. *Which city is the kid-friendly paradise?*



Show 10 entries

Search:

	city	count
1	Toronto	4911
2	Las Vegas	4709
3	Phoenix	3083
4	Montréal	2104
5	Charlotte	1908
6	Calgary	1825
7	Pittsburgh	1549
8	Mississauga	1094
9	Scottsdale	1044
10	Mesa	994

Showing 1 to 10 of 670 entries

Previous 1 2 3 4 5 … 67 Next

Fig 14. Cities that are kid-friendly

We can see that Toronto, Las Vegas and Phoenix are the most kid-friendly cities.

7. *Can we show an interactive map of restaurants in the city with an indication of their ratings?*

We chose to visualize all the restaurants in Las Vegas.

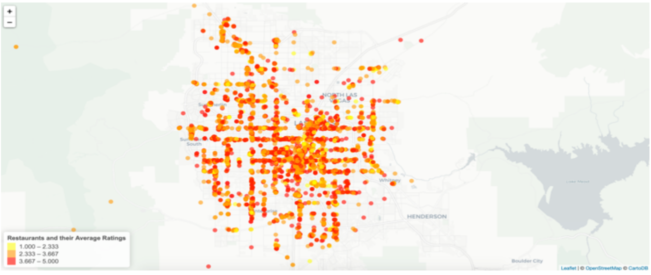


Fig 15. An interactive map of restaurants in Las Vegas and their average ratings indicated by gradient colors.

8. *What is the comparison between the maximum number of reviews for a restaurant and the number of high ratings (in Las Vegas)?*

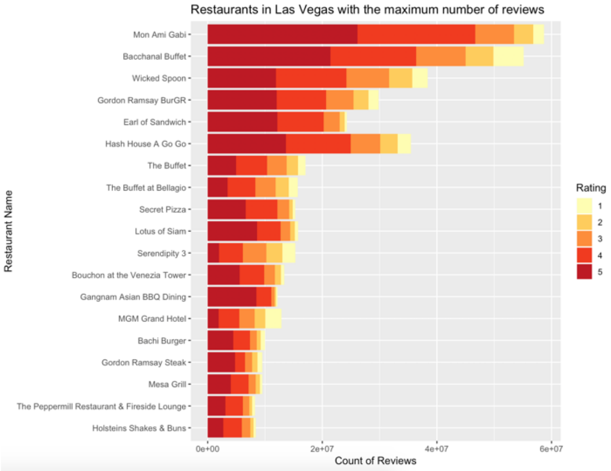
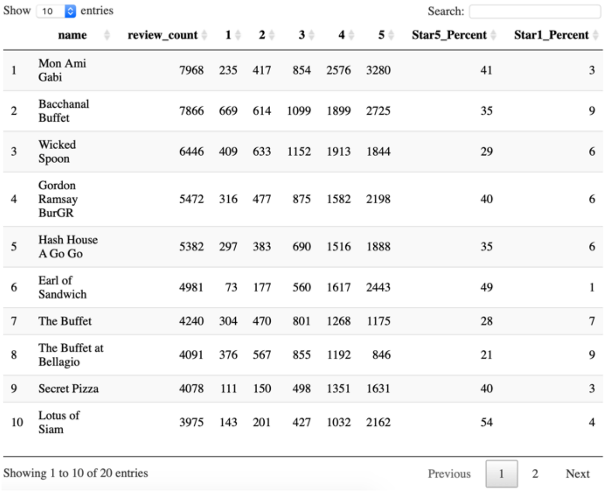


Fig 16. Restaurants in Las Vegas with maximum number of reviews and their corresponding ratings



Show 10 entries Search:

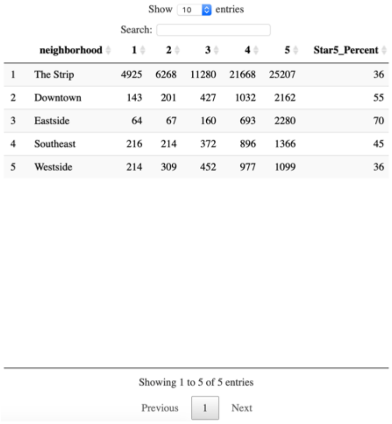
	name	review_count	1	2	3	4	5	Star5_Percent	Star1_Percent
1	Mon Ami Gabi	7968	235	417	854	2576	3280	41	3
2	Bacchanal Buffet	7866	669	614	1099	1899	2725	35	9
3	Wicked Spoon	6446	409	633	1152	1913	1844	29	6
4	Gordon Ramsay BurGR	5472	316	477	875	1582	2198	40	6
5	Hash House A Go Go	5382	297	383	690	1516	1888	35	6
6	Earl of Sandwich	4981	73	177	560	1617	2443	49	1
7	The Buffet	4240	304	470	801	1268	1175	28	7
8	The Buffet at Bellagio	4091	376	567	855	1192	846	21	9
9	Secret Pizza	4078	111	150	498	1351	1631	40	3
10	Lotus of Siam	3975	143	201	427	1032	2162	54	4

Showing 1 to 10 of 20 entries

Fig 17. List of restaurants with maximum reviews and their ratings

We can see that Mon Ami Gabi has the highest number of reviews. However, Lotus of Siam has a better rate of 5-star ratings. This shows that only the number of reviews does not provide us with reliable information about how good a restaurant is.

9. *Which neighborhood houses the maximum number of highly rated restaurants?*



Show 10 entries

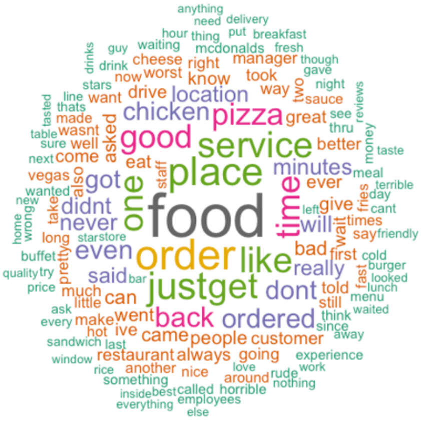
Search:

	neighborhood	1	2	3	4	5	Star5_Percent
1	The Strip	4925	6268	11280	21668	25207	36
2	Downtown	143	201	427	1032	2162	55
3	Eastside	64	67	160	693	2280	70
4	Southeast	216	214	372	896	1366	45
5	Westside	214	309	452	977	1099	36

Showing 1 to 5 of 5 entries

Fig 18. The top 5 neighborhood in Las Vegas with the highest number of highly rated restaurants

10. *Can we create a word cloud of the most frequently occurring phrases for the low rated restaurants?*



VI. Conclusions

The findings of our project are as follows:

1. The top-rated category of food is Pizza followed by Mexican, Chinese and Italian.
2. The users are more likely to focus on the positives rather than the negatives. If users are not happy with his dining experience, the users would prefer to not comment about it.
3. Expensive restaurants have a better rating.
4. Toronto has the maximum number of 5-star rated restaurants followed by Las Vegas and Montreal.
5. Bars, Cafes and Sandwich joints are the top categories of 5-star rated restaurants.
6. Toronto is the kid-friendly paradise followed by Las Vegas and Phoenix.
7. We cannot conclude the rating of a restaurant based on the number of reviews it has.
8. In Las Vegas, The Strip houses the maximum number of highly rated restaurants followed by Downtown and Eastside.

VII. Future Work

We tried to implement various supervised algorithms like Random Forest, SVM, but, were not able to accomplish the same due to the size of the dataset and hardware issues. As an extension to this project, in the future, we would like to reduce the size of the review dataset through dimensionality reduction and perform sentiment analysis along with supervised learnings.

References

- https://www.yelp.com/dataset/challenge
- https://arxiv.org/pdf/1605.05362.pdf
- https://blog.exploratory.io/working-with-json-data-in-very-simple-way-ad7ebcc0bb89
- https://rstudio-pubs-static.s3.amazonaws.com/228456_6a9ded5cd0324b87b9e13b1a5d1b4555.html
- https://www.kaggle.com/ambarish/a-very-extensive-data-analysis-of-yelp
- https://nycdatascience.com/blog/student-works/project-1-exploratory-visualizations-of-yelp-academic-dataset-draft/
- https://rpubs.com/JerryTsien/YelpReport
- https://github.com/piyushghai/Yelp-Dataset-Analysis
- https://www.r-bloggers.com/does-sentiment-analysis-work-a-tidy-analysis-of-yelp-reviews/
- https://www.statmethods.net/r-tutorial/index.html
- http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know
- https://rstudio.github.io/leaflet/
- https://www.statmethods.net/graphs/index.html