Big Data Project - Yelp Dataset

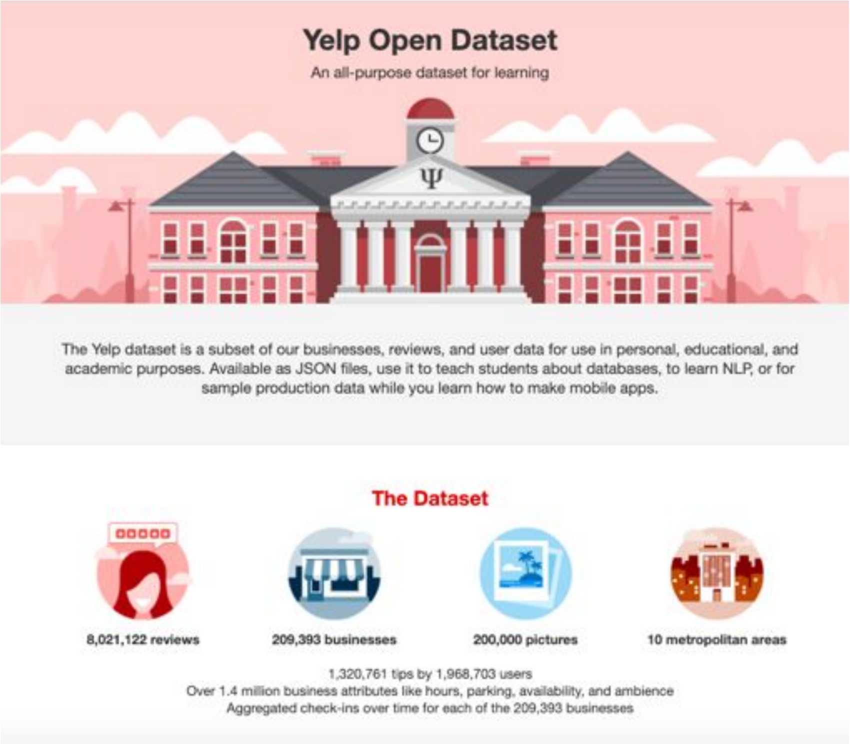
Data Aggregation and Recommendation System Implementation

Truc Cao

Tejinder Kaur

Amman Shareef

Kathia Teran



Problem and Motivation

The Restaurant market in New York is worth $17 billion dollars, with currently 31,061 businesses. Having a dataset like that of Yelp's readily available online is of much value to any business trying to enter this market.There is a myriad of ways a business can benefit from a dataset like this, since it has detailed information on what customers like and dislike (among other information) about the way in which a restaurant business operates.

Yelp dataset has a lot of data but hard to extract useful information

We want to:

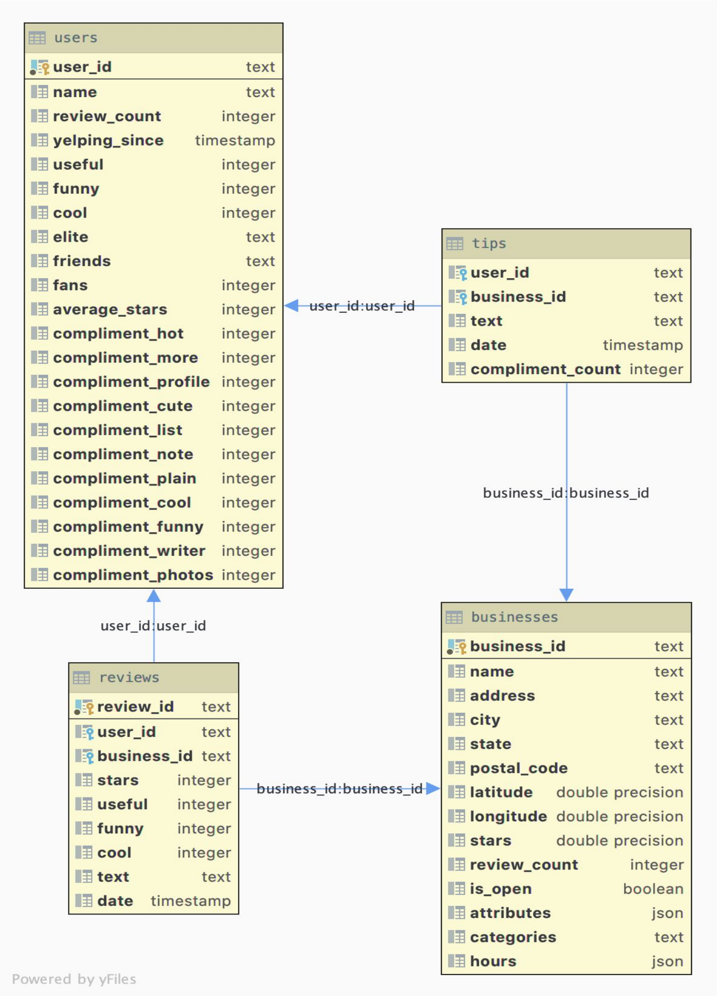
- Get statistics on restaurants (cuisines, popularity, …)
- Recommend restaurants to customers to increase revenue
- Recommend users to restaurants so they can target them with advertising

Using:

- Spark (Big Data Framework) to cleanup, filter and aggregate data
- Research papers about recommendation system
- Spark ML (Machine Learning) to provide useful recommendation

Yelp Dataset

- Yelp is a popular online crowd-sourced local business review platform.
- Sample: https://www.yelp.com/dataset
- The dataset contains:
  - 192,609 businesses
  - 1,637,138 users
  - 6,685,900 reviews
  - 1,223,094 tips
  - 161,950 check-ins
  - 36 states
  - 1,307 cities
  - All the data files are in JSON format.



Samples


```
{
  "business_id": "0W27hbZN7Z-PrkhAb0y9Eg",
  "name": "B Montréal",
  "address": "1207-A Rue Rachel E",
  "city": "Montréal",
  "state": "QC",
  "postal_code": "H2J 2J8",
  "latitude": 45.5266477836,
  "longitude": -73.5735161233,
  "stars": 4.5,
  "review_count": 3,
  "is_open": 1,
  "attributes": {
    "RestaurantsPriceRange2": "1",
    "RestaurantsTableService": "False",
    "RestaurantsTakeOut": "True",
    "OutdoorSeating": "True",
    "RestaurantsReservations": "False",
    "Alcohol": "u'none'",
    "BikeParking": "True",
    "WheelchairAccessible": "True",
    "Caters": "True",
    "HasTV": "False",
    "GoodForMeal": "{'dessert': False, 'latenight': False, 'lunch': False, 'dinner': False, 'brunch': False, 'breakfast': False}",
    "GoodForKids": "True",
    "RestaurantsDelivery": "True",
    "WiFi": "u'free'"
  },
  "categories": "Coffee & Tea, Food, Juice Bars & Smoothies, Delis, Restaurants, Sandwiches",
  "hours": {
    "Wednesday": "9:0-20:0",
    "Thursday": "9:0-20:0",
    "Friday": "9:0-20:0",
    "Saturday": "8:0-16:0"
  }
}
```


business.json


```
{
  "business_id": "-Lw8Ve0NLbR0djHGw2fMOA",
  "date": "2014-06-30 22:57:27, 2014-06-30 22:58:28, 2017-05-19 18:24:18"
}
```


checkin.json


```
{
  "user_id": "zHseuNq_3b246ZgzcY8BXA",
  "name": "Briana",
  "review_count": 44,
  "yelping_since": "2009-10-16 23:54:29",
  "useful": 40,
  "funny": 3,
  "cool": 12,
  "elite": "",
  "friends": "_SEBcjCwgneOV1VV_vESfQ, vtNHzdtfsxCCsbc_JUK8Cw, EPyRgySYsR365gAgF2r4Ww, 3mNk60ynkQYRNJGf3YAqiA, NfU0zDaTMEQ4-X9dbQWd9A, Xx
  "fans": 2,
  "average_stars": 4.07,
  "compliment_hot": 0,
  "compliment_more": 1,
  "compliment_profile": 0,
  "compliment_cute": 0,
  "compliment_list": 0,
  "compliment_note": 0,
  "compliment_plain": 0,
  "compliment_cool": 0,
  "compliment_funny": 0,
  "compliment_writer": 0,
  "compliment_photos": 0
}
```


user.json

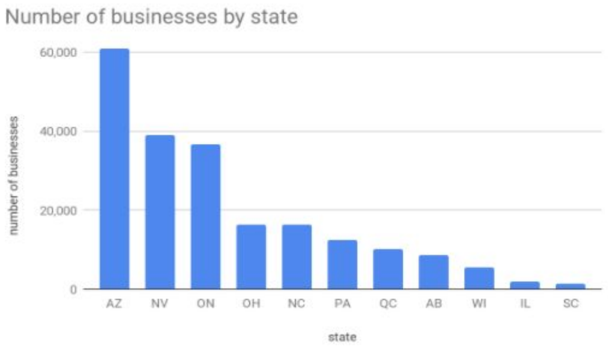

```
{
  "review_id": "BTDBNxb7m6wuSTy09_Zz4A",
  "user_id": "oV4PUFp4O2brd3bGhu5cjg",
  "business_id": "m97jaBYRscg-hqDjMVIIWg",
  "stars": 4,
  "useful": 0,
  "funny": 0,
  "cool": 0,
  "text": "This is our go-to place for lunch and just a friendly atmosphere. Very consistent food. A
  "date": "2017-03-03 22:35:42"
}
```

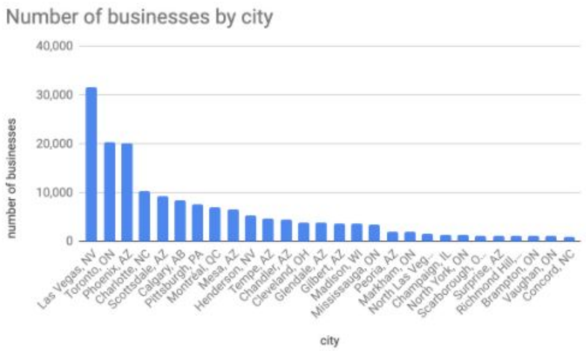

review.json

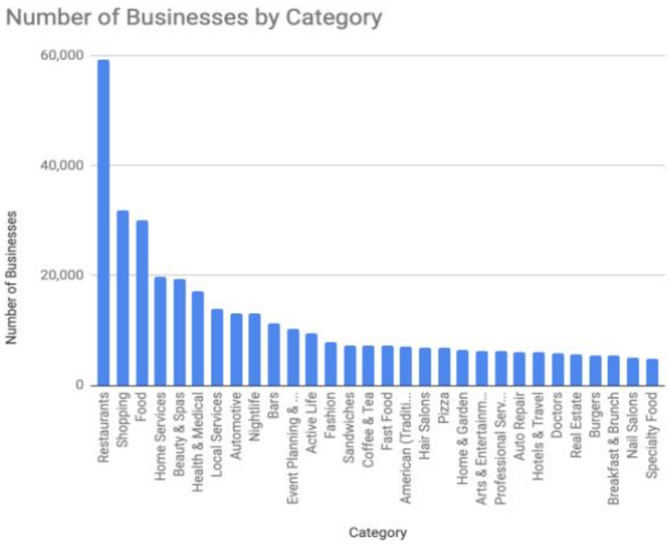

```
{
  "user_id": "ky3DB9i9lDJ7OAZdkZyv7g",
  "business_id": "m9ybLDUrbqgso1IT06bBLA",
  "text": "Excellent price on tires here!",
  "date": "2016-01-07 18:01:31",
  "compliment_count": 0
}
```


tip.json

Dataset: Distribution



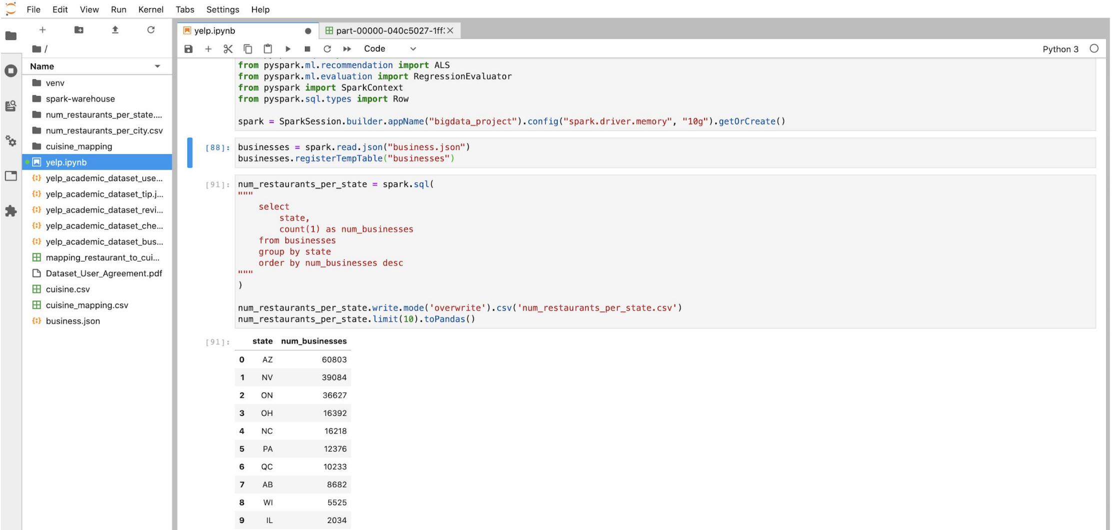




- Out of 36 states (US/Canada), 11 has more than 1,000 businesses
- Out of 1,307 cities, 28 has more than 1000 businesses
- Restaurant is by far the most represented category

Preparing the data

- Use Jupyter notebook and Spark SQL
- Jupyter notebook:
  - Interactive notebook in the browser
  - Can share easily with other people
- SparkSQL:
  - SQL queries are easier to write (and to read) than dataframe operations
  - Take advantage of Spark to scale to multiple machines to speed up the process



Preparing the data

Steps:

- Only keep restaurants: “Restaurants”
- Break down categories (one row per category):
  - “Ramen, Sushi” becomes 2 rows for each
- Recategorize using 32 kind of cuisines (American, Korean):
  - Use manually created mapping (114 mappings)
  - In case of multiple cuisine we choose one arbitrarily
- Only keep restaurants with a cuisine
- Only keep cities that have over 200 restaurants
- Only keep reviews and users associated to them

	cuisine	category
0	American	American (Traditional)
1	American	American (New)
2	American	Steakhouses
3	American	Bagels
4	American	Cajun/Creole
...	...	...
109	Asian	Cambodian
110	Korean	Korean
111	Asian	Laotian
112	Asian	BurmesePizza
113	Thai	Thai

cuisine.csv (Cuisine mapping)

Initialization

```
spark = SparkSession.builder.appName("yelp").config("spark.driver.memory", "10g").getOrCreate()
businesses = spark.read.json("yelp_academic_dataset_business.json")
businesses.registerTempTable("businesses")
```

	address	attributes	business_id	categories	city	hours	is_open	latitude	longitude	name
0	404 E Green St	(None, None, 'none', None, None, None, None, F...	pQeaRpvuhoEqudo3uymHIQ	Ethnic Food, Food Trucks, Specialty Food, Impo...	Champaign	(11:30-14:30, 11:30-14:30, None, None, 11:30-1...	1	40.110446	-88.233073	The Empanadas House
1	4508 E Independence Blvd	(None, None, None, None, None, None, None, Non...	CsLQLiRoafpJPJSkNX2h5Q	Food, Restaurants, Grocery, Middle Eastern	Charlotte	None	0	35.194894	-80.767442	Middle East Deli
2	15480 Bayview Avenue, unit D0110	(None, None, u'none', None, None, None, None, ...	eBEfgOPG7pvFhb2wcG9I7w	Restaurants, Cheesesteaks, Poutineries	Aurora	(11:0-22:0, 11:0-22:0, 11:0-22:0, 11:0-21:0, 1...	1	44.010962	-79.448677	Philthy Phillys

Break down categories:

```
restaurant_categories_df = spark.sql("""
    SELECT
        business_id,
        explode(split(categories, ', ')) as category
    FROM businesses
    where categories like '%Restaurants%'
""")
restaurant_categories_df.registerTempTable('restaurant_categories')
```

→

	business_id	category
0	AtD6B83S4Mbmq0t7iDnUVA	Sushi Bars
1	AtD6B83S4Mbmq0t7iDnUVA	Dim Sum
2	AtD6B83S4Mbmq0t7iDnUVA	Ramen

Map to cuisines:

```
cuisine_mapping_df = spark.read.csv("cuisine.csv", header=True)
cuisine_mapping_df.registerTempTable("cuisine_mapping")

restaurant_cuisines_df = spark.sql("""
    SELECT
        business_id,
        LAST(cuisine) as cuisine
    FROM restaurant_categories
    JOIN cuisine_mapping USING (category)
    GROUP BY business_id
""")
restaurant_cuisines_df.registerTempTable('restaurant_cuisines')
```

→

business_id	cuisine
--9e1ONYQuAa-CB_Rrw7Tw	American
-VAsjhmAbKF3Pb_-8rh3xg	Coffee & Tea
-cxD1NimFldATDUsN-oa3A	Mexican
-r8SvItXXG6_T3mP5GXRAw	Coffee & Tea
0859wfd1BQHG46Zpwhc0ZQ	Bars
09OYbFNrS1n8u5gE6W9ItA	German
0DwMrcy7_X_C_mP8_QcXug	American
0bqV9uzFVz98Bn_RlmcJTg	Mexican
0owIRP_z5RcYKKmh5RnD7A	Fast Food
1NmGVWYIF4iMngM6arKJTQ	Vietnamese

Final dataset

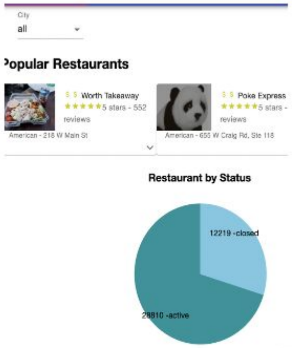
- 192,609 businesses
- 1,637,138 users
- 6,685,900 reviews
- 1,223,094 tips
- 161,950 check-ins
- 36 states
- 1,307 cities

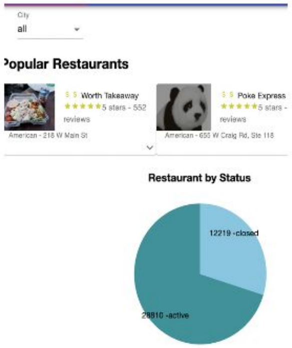
→

- 41,029 restaurants (-78%)
- 1,006,767 users (-39%)
- 3,447,322 reviews (-48%)
- 668,619 tips (-45%)
- 39,896 check-ins (-75%)
- 10 states -72%()
- 30 cities (-98%)

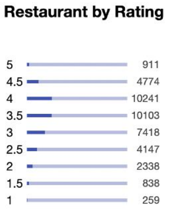
Computing different aggregations of the data

- Number of restaurants that are still active/closed for each dimension
- Number of restaurants /ratings / city
- Most popular restaurant and influencer for dimension (rating, cuisine, ...)
- Do all the aggregations using Spark SQL

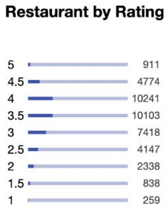




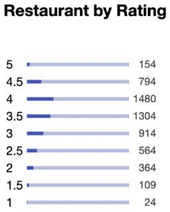
- All,



las vegas,



toronto



Timeseries data

Number or checkins / cuisine / city over time

Map tables from file

```
restaurants_df = spark.read.json("final_restaurants_all.json")
restaurants_df.registerTempTable('restaurants')

checkins_raw_df = spark.read.csv("final_restaurant_checkin_all.json")
checkins_raw_df.registerTempTable('checkins_raw')
```

Master aggregation by city/state, cuisine, month

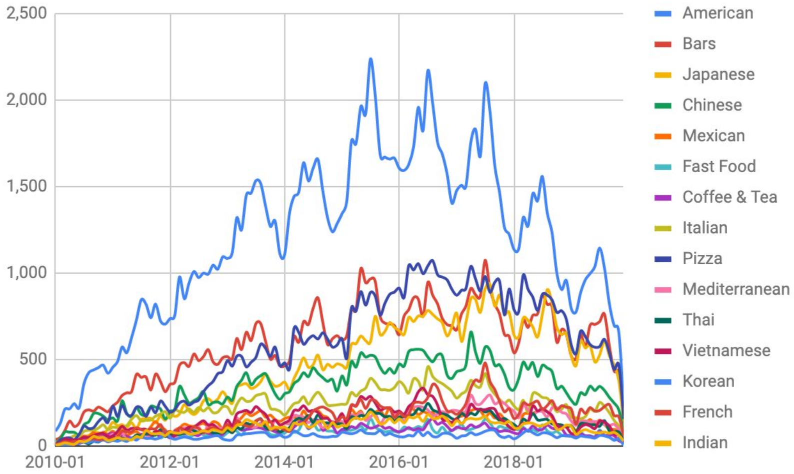
```
checkins_by_date_df = spark.sql("""
    WITH checkin_date AS (
        SELECT
            business_id,
            explode(split(date, ', ')) as date
        FROM checkins
    )
    SELECT
        city,
        state,
        cuisine,
        substring(date, 0, 7) as date,
        count(1) as num_checkins
    FROM checkin_date
    JOIN restaurants USING (business_id)
    GROUP BY city, state, cuisine, date
    ORDER BY date
""")
```

Pivot table for number of checkins for each cuisine in Toronto over time

```
popular_cuisines = ['American', 'Bars', 'Japanese', 'Chinese', 'Mexican', 'Fast Food', 'Coffee & Tea', 'Italian', 'Pizza', 'Mediterranean', 'Thai',
                    'Vietnamese', 'Korean', 'French', 'Indian']
tmp_checkins_cuisine_toronto_df = checkins_by_date_df.filter(f.col('city') == 'Toronto').filter(f.col('cuisine').isin(popular_cuisines)) \
                                                     .groupBy('date').pivot('cuisine').sum('num_checkins').sort('date')
checkins_cuisine_toronto_df = tmp_checkins_cuisine_toronto_df.na.fill(0)
checkins_cuisine_toronto_df.repartition(1).write.mode('overwrite').csv('checkins_cuisine_toronto.csv')
```

```
checkins_cuisine_toronto_df.toPandas()
```

	date	American	Bars	Chinese	Coffee & Tea	Fast Food	French	Indian	Italian	Japanese	Korean	Mediterranean	Mexican	Pizza	Thai	Vietnamese
0	2010-01	86	37	8	22	3	4	6	19	8	2	4	14	5	9	4
1	2010-02	132	50	27	41	4	8	19	32	35	6	5	18	6	17	10
2	2010-03	212	88	32	81	21	10	14	39	45	10	6	40	3	25	18
3	2010-04	215	144	35	80	8	20	16	45	47	11	9	36	7	22	13
4	2010-05	245	113	30	79	18	17	16	60	37	9	12	35	13	27	10



Number of checkins by cuisine over time in Toronto

Influencers: how top reviewers rate businesses

•Influencers being those users with the highest amount of: friends, fans, useful rating, elite status, and oldest accounts (‘yelping since’).

```
spark = SparkSession.builder.appName("Popular_Reviewer").getOrCreate()

# Get the raw data CONVERTS JSON, used to be 'lines'
business_dataFrame = spark.read.json(path1)
reviews_dataFrame = spark.read.json(path2)
users_dataFrame = spark.read.json(path3)

# Some SQL-style magic to sort all movies by popularity in one line!
top_business = business_dataFrame.select("business_id", "name", "city").orderBy("city", ascending=False)
top_reviews = reviews_dataFrame.select("business_id", "user_id").orderBy("user_id", ascending=False)
top_reviewer = users_dataFrame.select("user_id", "name", "review_count").orderBy("review_count", ascending=False)

top_reviewers = top_business.join(top_reviews, on=['business_id'], how='inner').orderBy("city", ascending=False).\
                    join(top_reviewer, on=['user_id'], how='inner').orderBy("review_count", ascending=False)

top_reviewers.show(10, False)
```

Most Popular Influencers

I. BASED ON CITY (considering Las Vegas as the most yelped city for restaurants)

- Restaurant and city with the highest review count:

business_id	name	city	review_count
4JNXUYY8wbaaDmk3BPzlWw	Mon Ami Gabi	Las Vegas	8348
f4x1YBxkLrZg652xt2KR5g	Hash House A Go Go	Las Vegas	5763
cYwJA2A6I12KNkm2rtXd5g	Gordon Ramsay BurGR	Las Vegas	5484

TOP

- **'Useful'** influencer in Las Vegas (review count is from the business)

user_id	business_id	name	city	review_count	name	useful
--2vR0DIsmQ6WfcSzKWigw	IB8zLlGraOg9LU7qQVLPyg	Fashion Show	Las Vegas	739	Harald	154202

- 'Oldest **(yelping since)**' influencer in Las Vegas (business review count)

user_id	business_id	name	city	review_count	name	yelping_since
nkN_do3fJ9xekchVC-v68A	ubz4CaZXagQuGv2N9gFAdw	Botero	Las Vegas	434	Jeremy	2004-10-12 08:46:43

- Influencer with the most **'Fans'** in Las Vegas (business review count)

user_id	business_id	name	city	review_count	name	fans
37cpUoM8hlkSQfReIEBd-Q	0qet57CmMA5qUm6gPFUTpg	Di Fara Pizza	Las Vegas	150	Mike	9538

- Influencer with the most **'review counts'** in Las Vegas

user_id	business_id	name	city	name	review_count
8k3aO-mPeyhbR5HUucA5aA	6Q7-wkCPc1KF75jZLOTcMw	Circus Circus Las Vegas Hotel and Casino	Las Vegas	Victor	13278

II. BASED ON CATEGORY:

- Influencer with the most **'Fans'**

user_id	business_id	name	city	categories	name	fans
37cpUoM8hlkSQfReIEBd-Q	lYCeqldIiOggsbByH3RRhw	Di Fara Pizza	Las Vegas	Restaurants, Italian	Mike	9538

- 'Oldest (**yelping since**)' influencer

user_id	business_id	name	city	categories	name	yelping_since
c6HT44PKCaXqzN_BdgKPCw	u8C8pRvaHXg3PgDrsUHJHQ	Papa Del's Pizza	Champaign	Food Delivery Ser...	Russel	2004-10-12 08:40:43

- Influencer with the most **'review counts'**

user_id	business_id	name	city	categories	name	review_count
8k3aO-mPeyhbR5HUucA5aA	6Q7-wkCPc1KF75jZLOTcMw	Circus Circus …	Las Vegas	Arts & Entertainment, Restaurants	Victor	13278
RtGqdDBvvBCjcu5dUqwfzA	oUX2bYbqjqST-urKbOHG6w	Loftti Cafe	Las Vegas	Desserts, Juice Bars & Smoothies…	Shila	12390

- Most **'Useful'** influencer in Las Vegas

user_id	business_id	name	city	categories	name	useful
--2vR0DIsmQ6WfcSzKWigw	uanCi4OGc1mHLGl_AT4JhQ	Treasure Island	Las Vegas	Hair Salons, Arts...	Harald	154202

III. Cities reviewed by the strongest influencers

- Las Vegas on top:

user_id	business_id	name	city	name	review_count
8k3aO-mPeyhbR5HUucA5aA	6Q7-wkCPc1KF75jZLOTcMw	Circus Circus Las Vegas Hotel and Casino	Las Vegas	Victor	13278
RtGqdDBvvBCjcu5dUqwfzA	oUX2bYbqjqST-urKbOHG6w	Loftti Cafe	Las Vegas	Shila	12390

Most Popular Influencers

- Las Vegas has the Influencer with the most **'Useful'** reviews (business review count)

user_id	business_id	name	city	review_count	name	useful
--2vR0DIsmQ6WfcSzKWigw	IB8zLlGraOg9LU7qQVLPyg	Fashion Show	Las Vegas	739	Harald	154202
--2vR0DIsmQ6WfcSzKWigw	uanCi4OGc1mHLGl_AT4JhQ	Treasure Island	Las Vegas	2487	Harald	154202
--2vR0DIsmQ6WfcSzKWigw	7dHYudt6OOIjiaxkSvv3lQ	In-N-Out Burger	Las Vegas	417	Harald	154202

- Champaign has the 'oldest (**yelping since**)' influencer (business review count)

user_id	business_id	name	city	review_count	name	yelping_since
c6HT44PKCaXqzN_BdgKPCw	u8C8pRvaHXg3PgDrsUHJHQ	Papa Del's Pizza	Champaign	402	Russel	2004-10-12 08:40:43

- Las Vegas has the influencer with the most **'fans'** (business review count)

user_id	business_id	name	city	review_count	name	fans
37cpUoM8hlkSQfReIEBd-Q	0qet57CmMA5qUm6gPFUTpg	Di Fara Pizza	Las Vegas	150	Mike	9538

V. Top influencers:

- The most **'Useful'**

user_id	name	useful
--2vR0DIsmQ6WfcSzKWigw	Harald	154202
JjXuiru1_ONzDkYVrHN0aw	Richard	99162
W7DHyQlY_kXls2iXt-_2Ag	Maggie	89792
Hi10sGSZNxQH3NLyWSZ1oA	Fox	89418
ax7SnXOTIpatbsmqHLqVow	Rohlin	81003

- The most **'review counts'**

user_id	name	review_count	average_stars
8k3aO-mPeyhbR5HUucA5aA	Victor	13278	3.28

- The most **'fans'**

user_id	name	fans
37cpUoM8hlkSQfReIEBd-Q	Mike	9538
hizGc5W1tBHPghM5YKCAtg	Katie	2964
eKUGKQRE-Ywi5dY55_zChg	Cherylynn	2434
iLjMdZi0Tm7DQxX1C1_2dg	Ruggy	2383
j14WgRoU_-2ZE1aw1dXrJg	Daniel	2132

- The oldest **'yelping since'**

user_id	name	yelping_since
c6HT44PKCaXqzN_BdgKPCw	Russel	2004-10-12 08:40:43
nkN_do3fJ9xekchVC-v68A	Jeremy	2004-10-12 08:46:43
wqoXYLWmpkEH0YvTmHBsJQ	Michael	2004-10-12 08:51:07
sE3ge33huDcNJGW3V4obww	Ken	2004-10-12 09:16:01
5iOHz6pHmXi9SoB5qomRWQ	Nader	2004-10-12 17:42:24

- Most recent **Elite** influencer

user_id	business_id	name	city	name	elite
3Fmj7MfGfsUUK1kTWCSL_g	D5oLn4j7eezCAoOsuYr8jA	ND Sushi & Grill	Toronto	Matthew	2018

Recommendation system

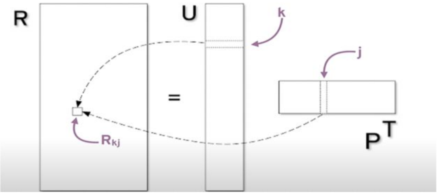
We want to recommend restaurants to users

- 2 main type of recommenders:
  - Content-based recommendations (good to recommend when user has no history)
  - Collaborative filtering (good when user has a history)
- Yelp Dataset:
  - Restaurant data contains very limited information
  - Very few reviews compared to the number of restaurants and user(sparse matrix)
- Chosen approach:
  - we use collaborative filtering through matrix factorization to predict user ratings based on past ratings
  - Use ALS in SparkML

ALS algorithm:

- Factorize a ratings matrix R into two latent factor matrices which when multiplied back, will give a approximation of the original ratings matrix. In the approximation matrix, all the cells will be filled by an estimated rating.
- if we want to predict how user K might rate product J we just multiply those two vector together
- Cost function:

$$\min_{x_\star, y_\star} \sum_{r_{u,i} \text{ is known}} (r_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_i\|^2)$$

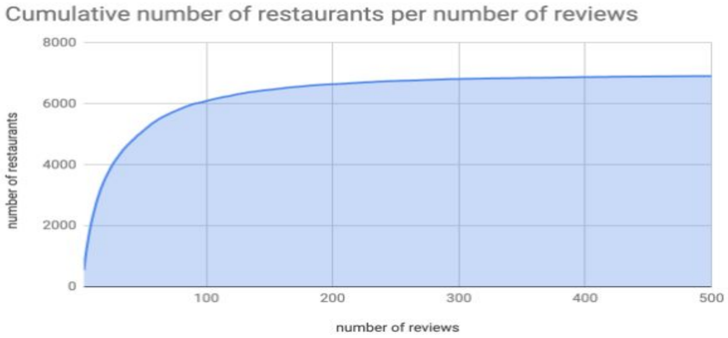


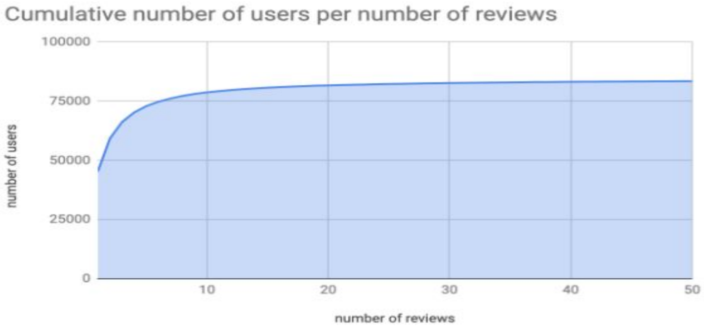
- It’s alternating because the process that generates those matrices U and P is done first by fixing U optimizing for P and then fixing P and optimizing for U and we repeat that process alternately.→ effective performance
- Using ASL-WR to avoid overfitting

Cleaning up the data even more

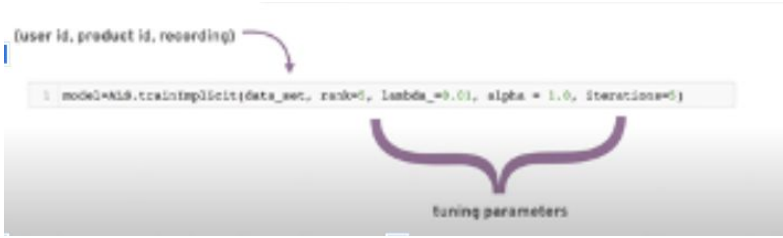
Reduce data to reduce memory consumption and speed up the process:

- Consider Toronto since it has the most number of restaurants
- Only keep restaurants with over 100 reviews
- Only keep users with over 10 reviews
- 6900 restaurant ---> 895 restaurants
- 80000 users ---> 5458 users





Tuning the recommender



ALS relies on 3 hyperparameters:

- Number of latent features
- Number of iterations
- Lambda of regularization

The best values of the hyperameters are the one which minimize the RSE cost function.
We run 420 experiments with different hyperparameters in 5 hours.

The best set of parameter is ….[show map]

Experiment result

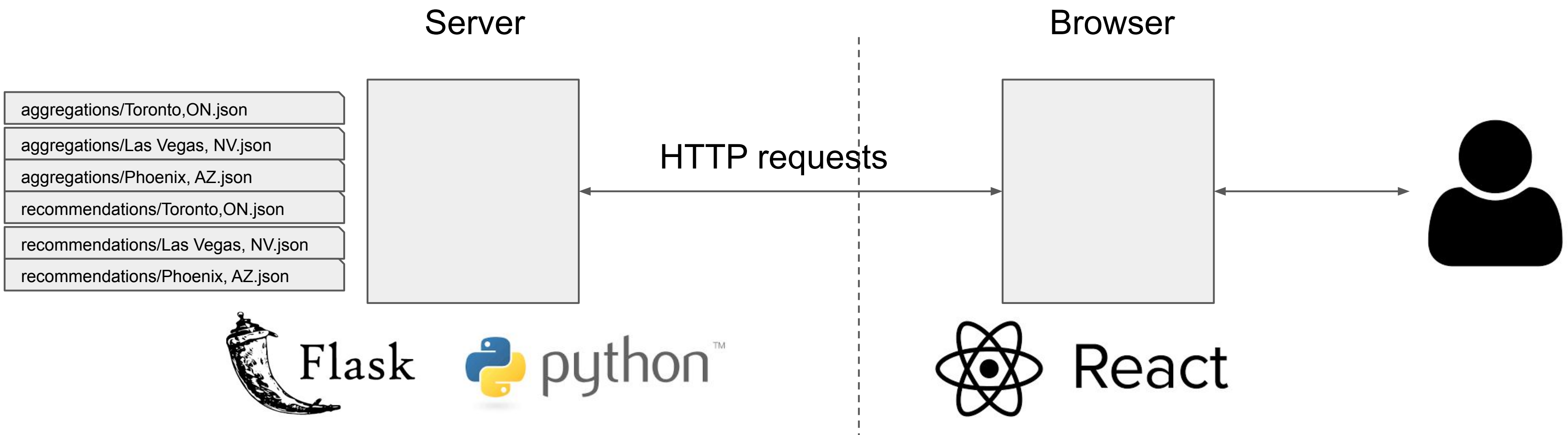
Running in…. Mins

RMSE = ….

Sample of recommendation

Demo

- Front-end:
  - React Framework: Popular Javascript Framework
  - Styling: Materialize
  - Map Library: Leaflet
  - Chart Library: VIS
- Backend:
  - Flask: Popular Python Framework for Web Services



Contribution

Amaan: preprocessing data using Spark SQL

TJ: aggregation data using Spark SQL

Kathia: aggregation data using Spark Core (low level API)

Truc: Recommendation with Spark ML

All: Web application

Future works

- Try with larger data using a large cluster of machines
- Show recommendation to real users and evaluate performance using A/B testing
- Try other models: hybrid, neural network
- Use database as backend

Questions?