Capstone Presentation

A Hybrid Recommendation System with Yelp Challenge Data

Team Yelper_Helper

S. Kickham, S. O'Mullane, R. Rad, A. Rubino, C. Shi

NYC Data Science Academy Cohort 9

06/20/2017

Outline

- Motivation
- Project Summary
- Data Source
- Data Preparation and Exploratory Analysis
- Hybrid Recommendation System
  - Natural Language Processing (NLP)
  - Collaborative Filtering (CF)
  - Social Network
  - Location-based
- Data Pipeline
  - Rec Engine
  - App Workflow (Flask - Kafka - Rec Engine - Kafka - Flask)
- Live Demo
- Lessons Learned and Future Work
- Acknowledgement

Background

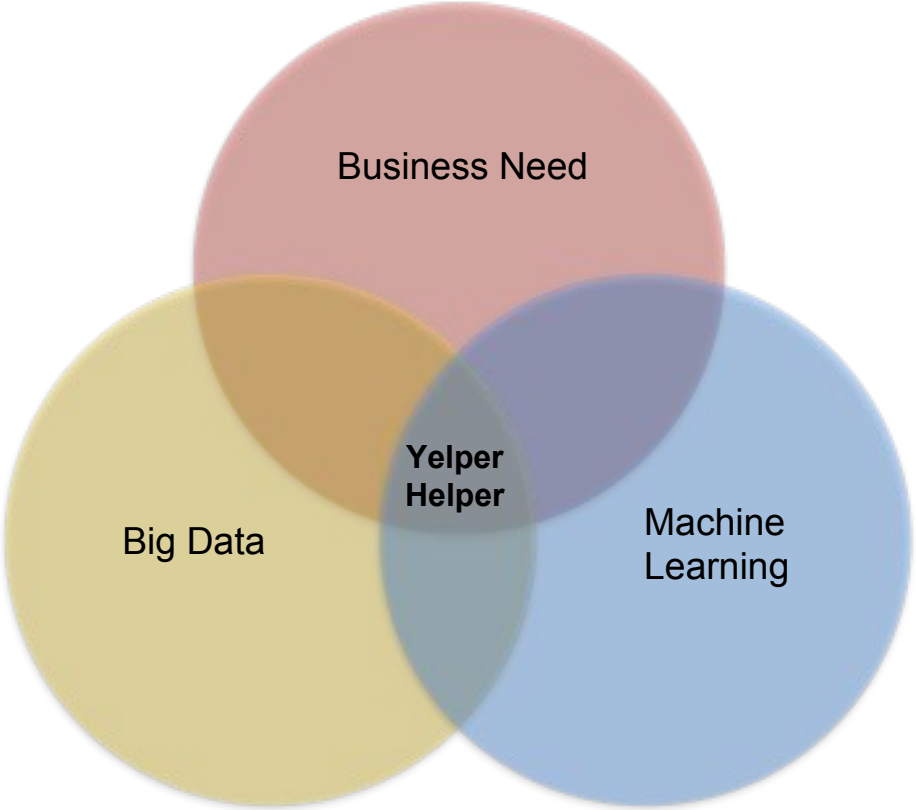
EDA

Recommendation Algorithms

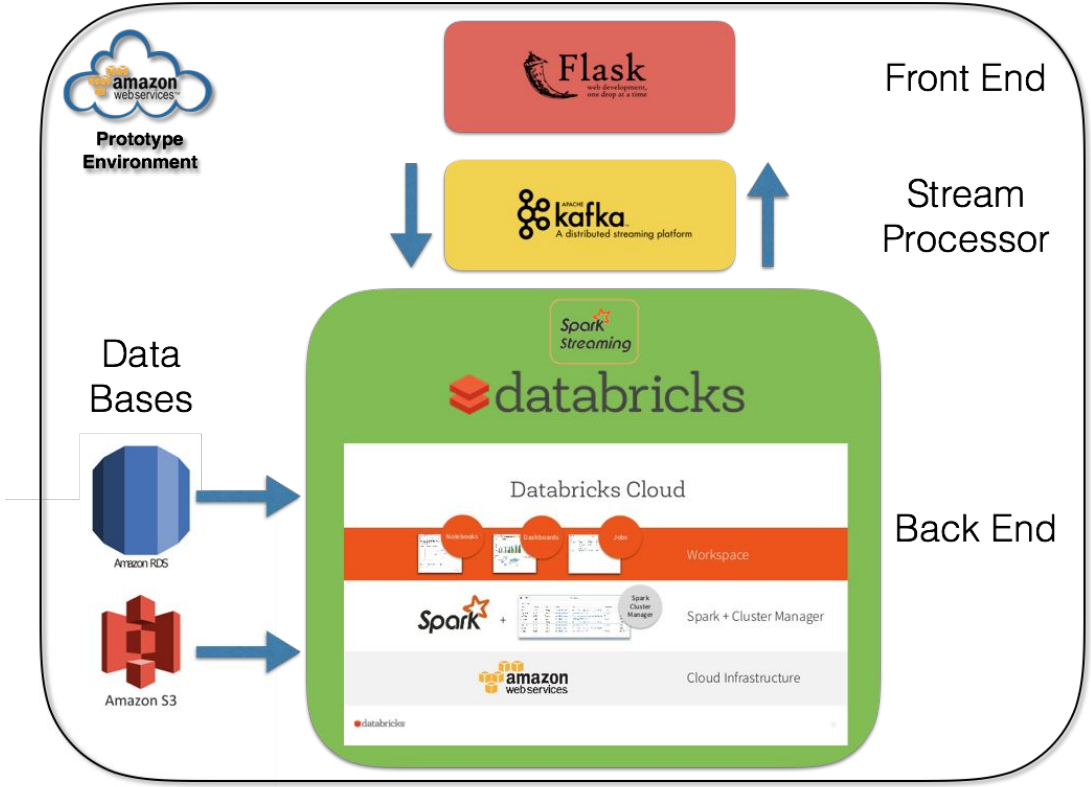
App Pipeline

Demo

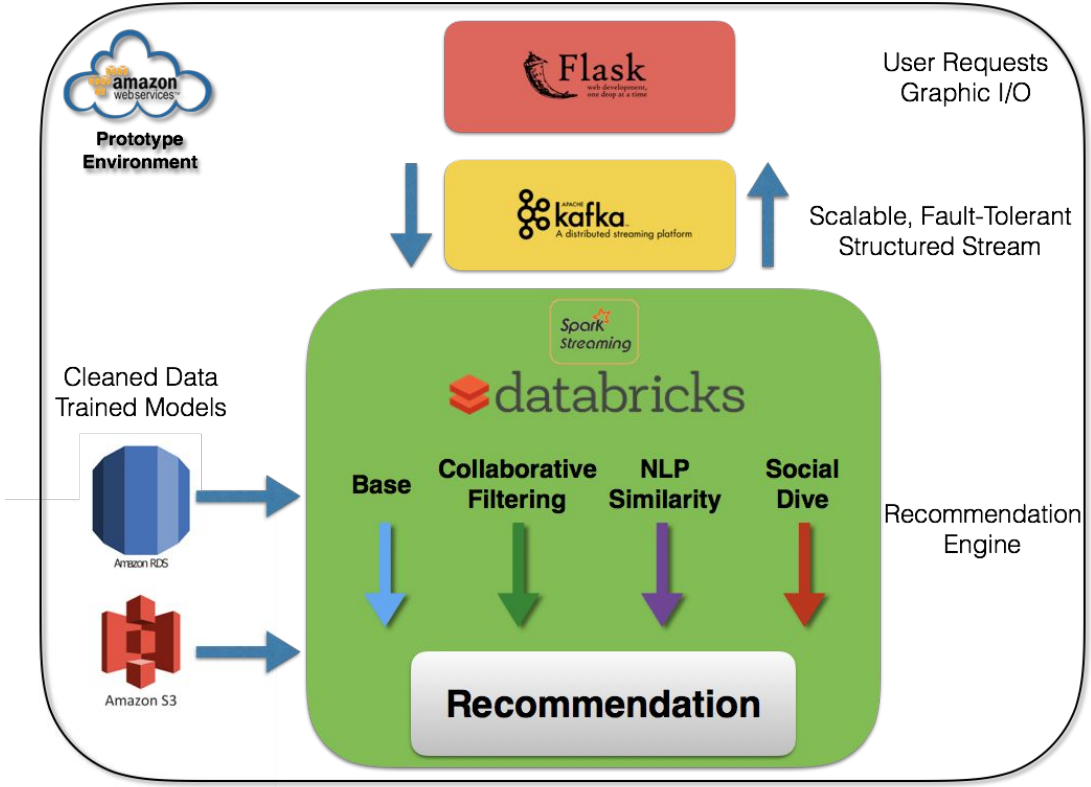
Motivation



Project Architecture



Project Architecture



Data Source

Yelp Dataset Challenge

Round 9 Of The Yelp Dataset Challenge: Our Largest Yet!

The Challenge Dataset:

- **4.1M** reviews and **947K** tips by **1M** users for **144K** businesses
- **1.1M** business attributes, e.g., hours, parking availability, ambience.
- Aggregated check-ins over time for each of the **125K** businesses
- **200,000** pictures from the included businesses

Get the Data

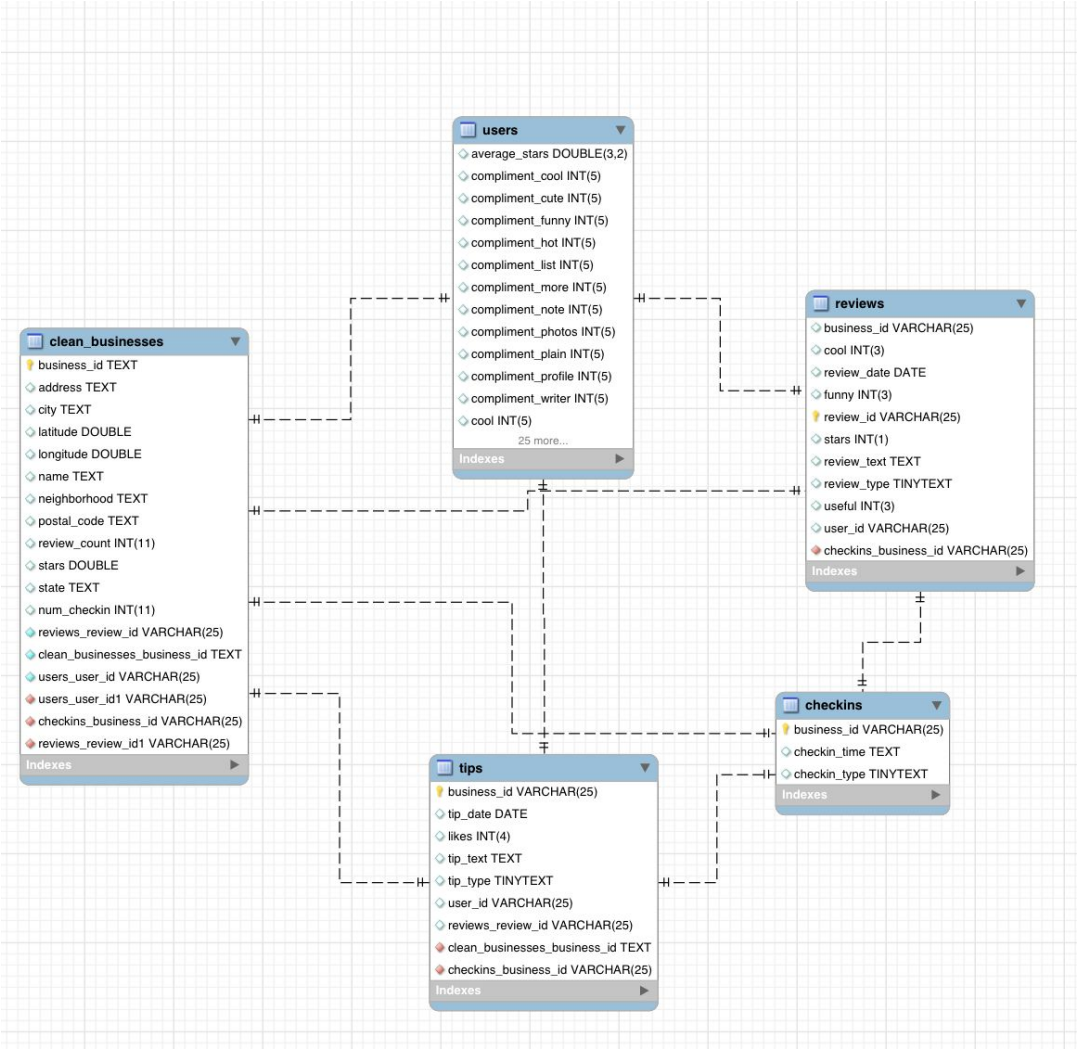
5 Gb text data
json format

Amazon RDS - MySQL Database

- RDS = Relational Database Service
- Main tables include:
  - **Reviews**
  - **Tips**
  - **Businesses**
  - **Check-ins**
  - **Users**
- The main tables allow for simple subtable creation, which allows for fast requests and data loads.

MySQL Schema Viz

- Primary Keys for all tables:
  - Businesses: business_id
  - Users: user_id
  - Reviews: review_id
  - Tips: user_id
  - Checkins: business_id
- RDS allowed us to create relationships between tables.



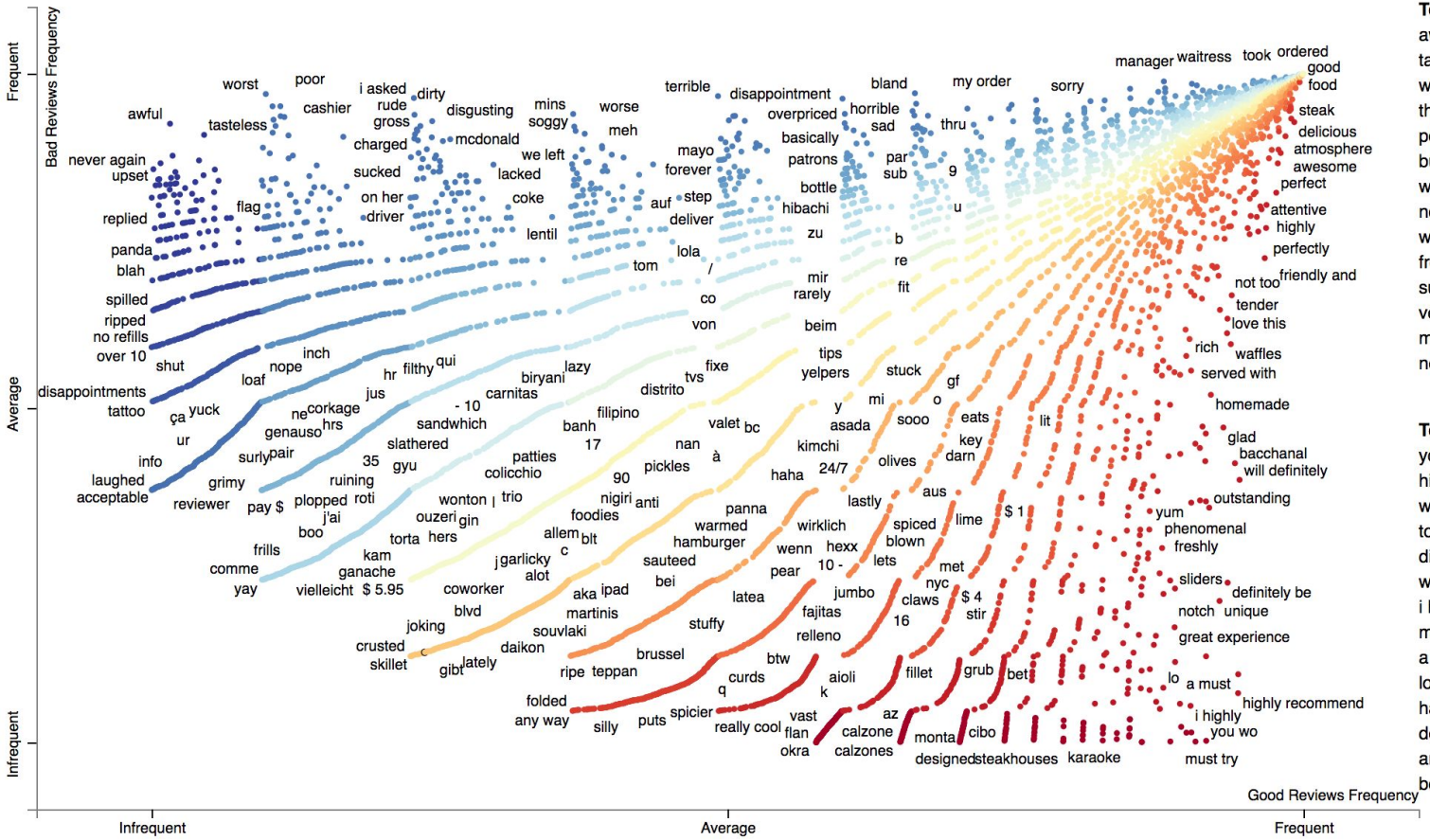
Read/Write Speed

Monitoring

	CURRENT VALUE	THRESHOLD	LAST HOUR
CPU	1%		
Memory	35.2 MB		
Storage	11,500 MB		

	CURRENT VALUE	LAST HOUR
Read IOPS	0/sec	
Write IOPS	0.367/sec	
Swap Usage	30 MB	

Word Frequency per Review Type



Top Bad Reviews
awful
tasteless
worst
the worst
poor
burnt
we asked
not even
will never
frozen
sucks
very disappointed
management
never again

Top Good Reviews
you wo
highly recommend
would definitely
to die
die for
well worth
i highly
must try
a must
loved it
hands down
definitely recommend
are amazing
boba

Characteristic
ramen
bacchanal
waitress
tacos
yelp
tasted
waiter
fries
wagyu
frites
flavorful
sashimi
overcooked
ribeye
groupon
ayce
craftsteak
filet
flavorless
guac
soggy
buffets
poutine
brisket
crispy
sushi
calamari
hibachi
appetizer
overpriced

Bad Reviews document count: 1,000; word count: 143,367
Good Reviews document count: 1,000; word count: 112,809

Sentiment Analysis

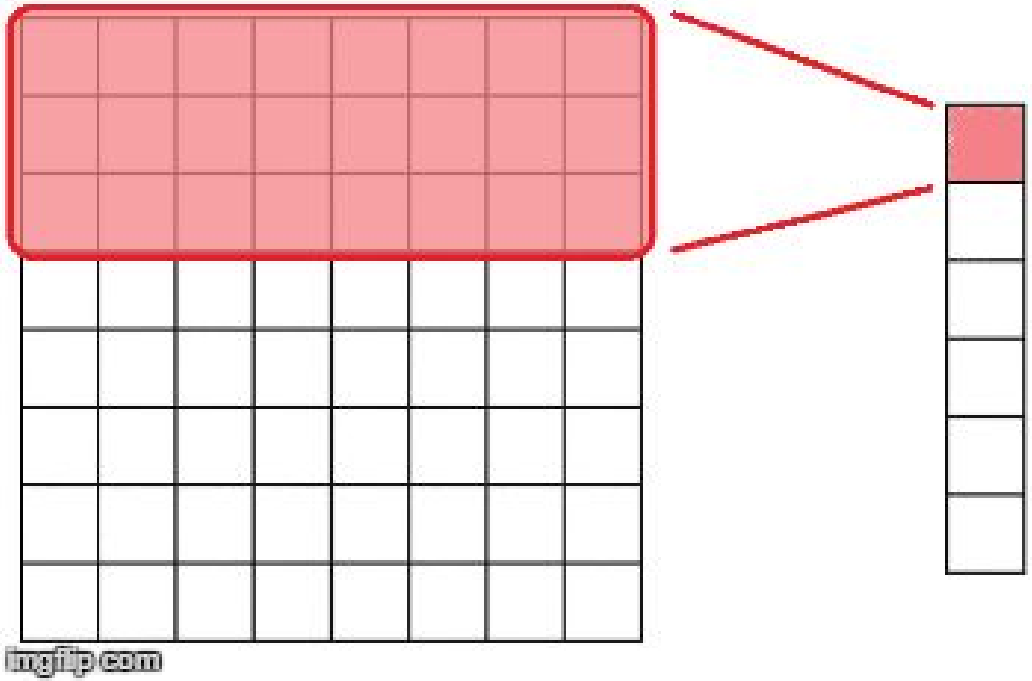
1. Decode
2. Binarize
3. Tokenize

1. Train/Test Split
2. Convolution
3. Pooling

1. Results
2. Spot Check
3. Base Rec.

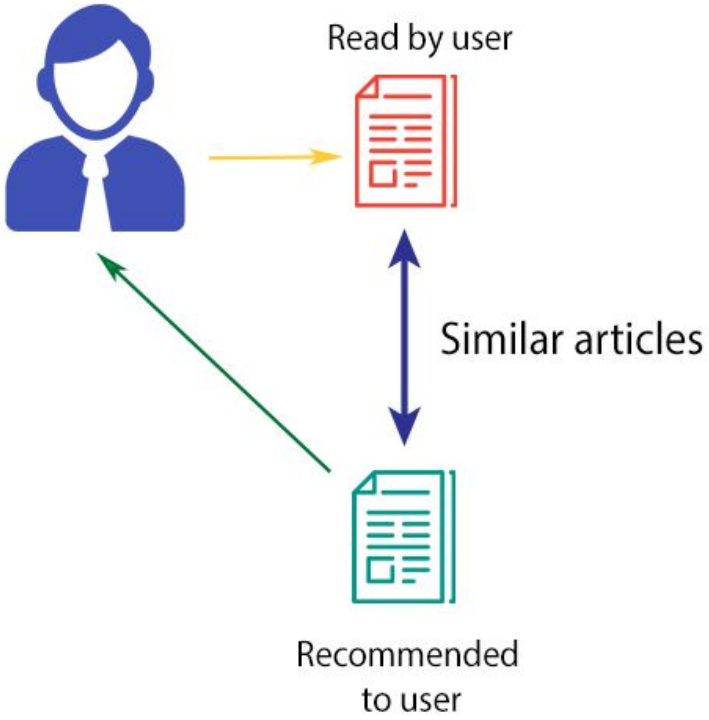
bit.ly/sent_analysis

Convolution and Pooling



Content-Based Recommendations

- Item-to-item based on user profile
- Keyword soft filtering
- Average Word2Vec



http://i65.tinypic.com/2ebah6c.png

Cosine Similarity using Word2Vec

Pre-Processing → **Neural Network** → **Similarity**

Pre-Processing

1. Tokenize
2. Stop Word Removal

Neural Network

1. Unsupervised
2. Skip-gram
3. Hyperparameters

Similarity

1. Cosine Similarity
2. Rank

bit.ly/word_2_vec

Word Similarity

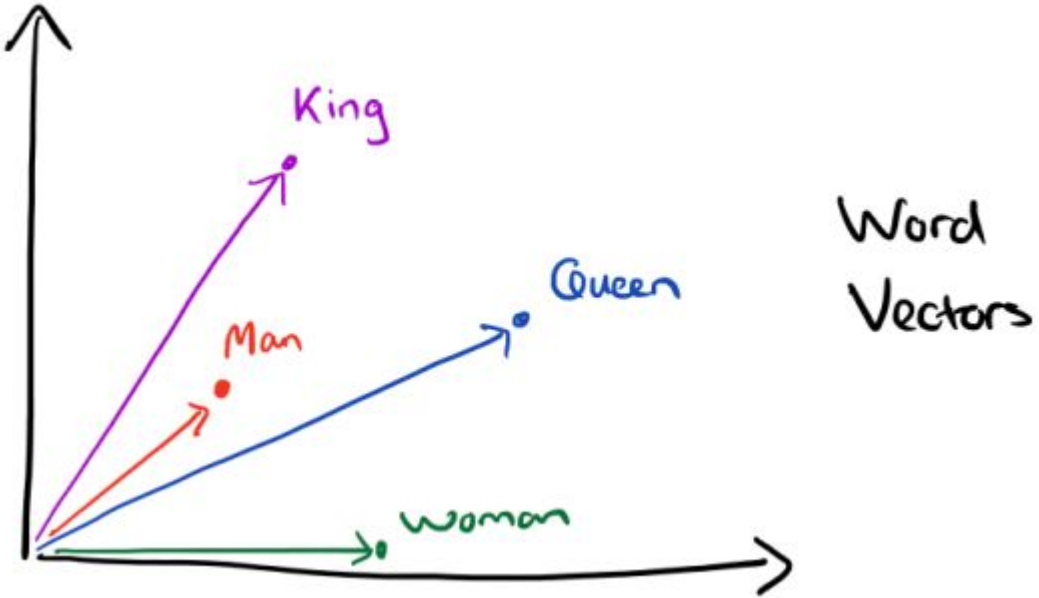
Cmd 28

```
#test similarity between words
synonyms = model.findSynonyms("crazy", 5)
synonyms.show(5)
```

```
+------------+------------------+
|        word|        similarity|
+------------+------------------+
|      insane| 0.585583245392664|
|        mean| 0.570725045955314|
|    weekends|0.5407901942365788|
|       cause|0.5246057760341053|
|ridiculously|0.5229965692070889|
+------------+------------------+
```

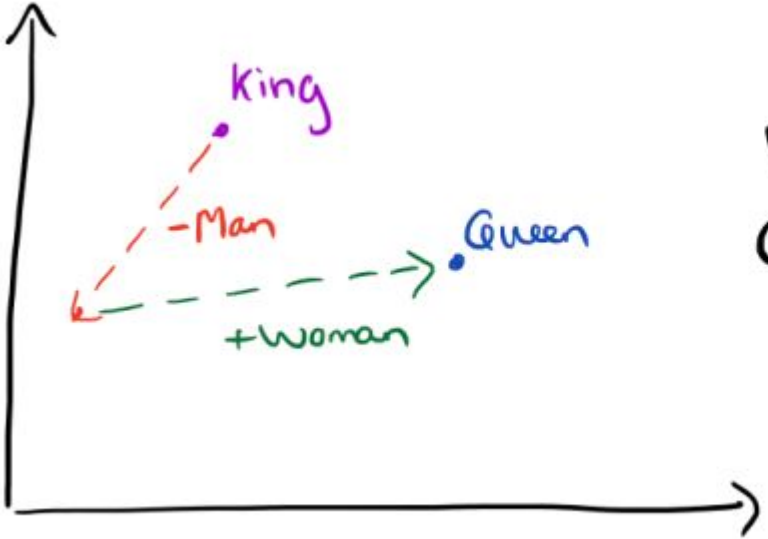
Command took 0.48 seconds -- by skickham@gmail.com at 6/15/2017, 2:59:40 PM on My Cluster

Word Algebra



https://adriancolyer.files.wordpress.com/2016/04/word2vec-king-queen-vectors.png?w=600

Word Algebra



Vector Composition

https://adriancolyer.files.wordpress.com/2016/04/word2vec-king-queen-composition.png?w=600

Word Algebra

https://adriancolyer.files.wordpress.com/2016/04/word2vec-king-queen-composition.png?w=600

Word Algebra

Word Algebra (visual?) (latex equation)

```
word_algebra(add=[u'filet_mignon', u'seafood'], subtract=[u'beef'])
```

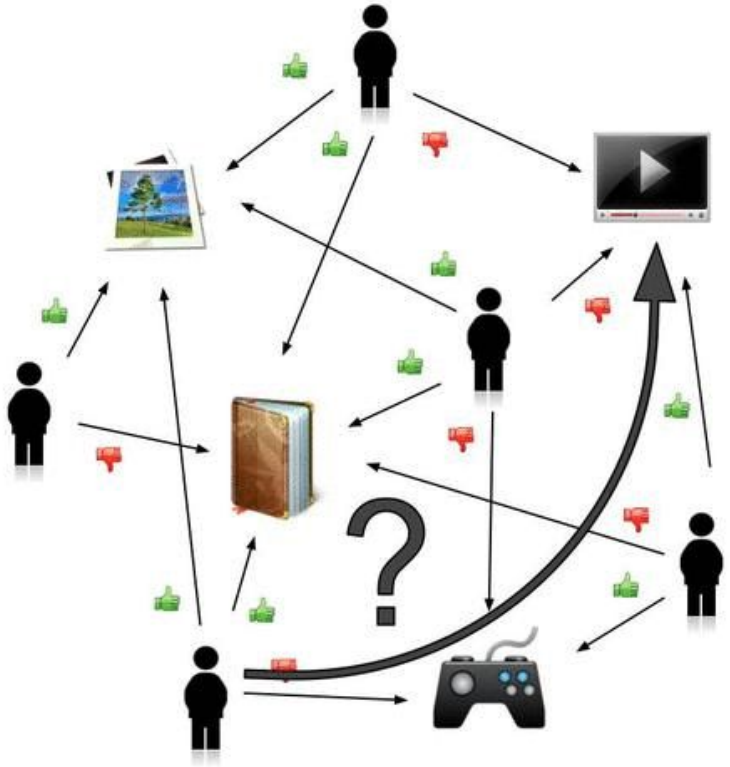
```
word_algebra(add=[u'fork', u'soup'])
```

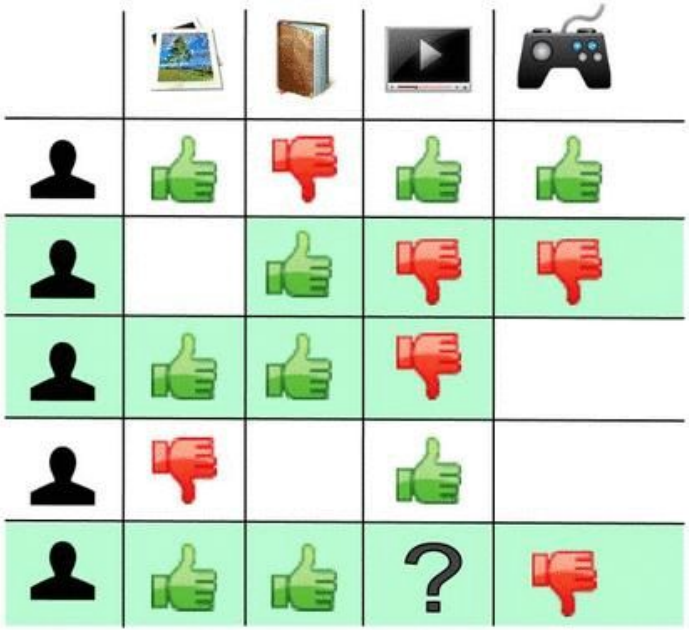
```
word_algebra(add=['drink', 'barley'])
```

NLP Summary (easy visual/flow)

- Sentiment Analysis as feature in Base Recommendation
- Cosine Similarity for Content-Based Recommendation

Collaborative Filtering 1

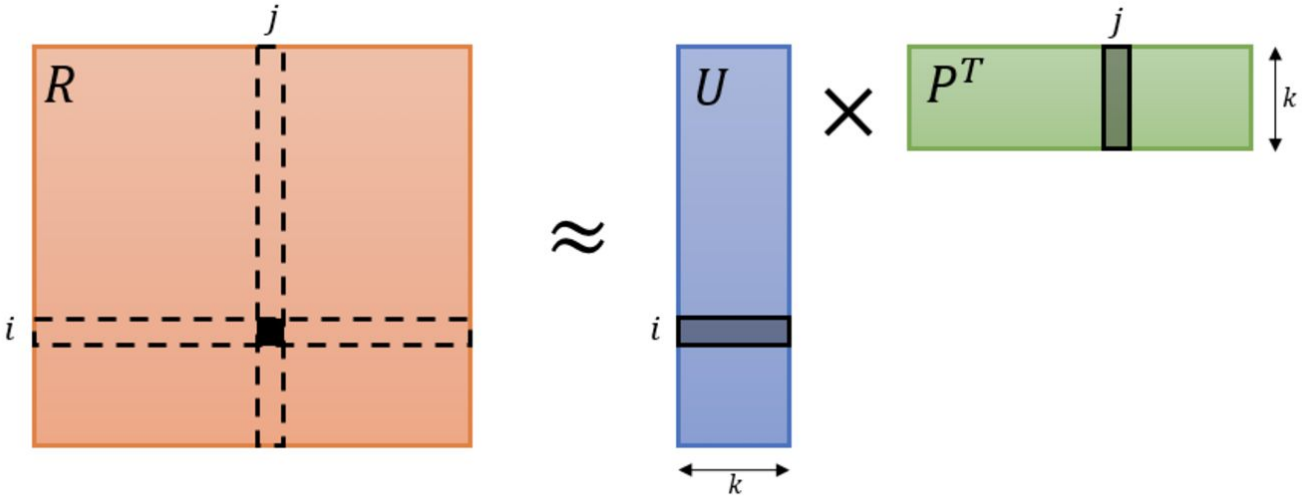




https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering 2

Matrix Factorization



Cost Function

$$J = ||R - U \times P^T||_2 + \lambda\left(||U||_2 + ||P||_2\right)$$

Yelp Social Network 1

nodeDF

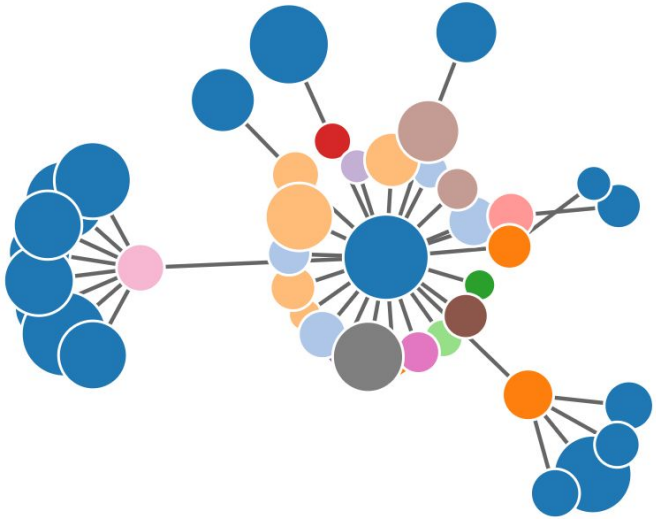
yelp_academic_dataset_user.json

```
{
    "user_id":"encrypted user id",
    "name":"first name",
    "review_count":number of reviews,
    "yelping_since": date formatted like "2009-12-19",
    "friends":["an array of encrypted ids of friends"],
    "useful":"number of useful votes sent by the user",
    "funny":"number of funny votes sent by the user",
    "cool":"number of cool votes sent by the user",
    "fans":"number of fans the user has",
```

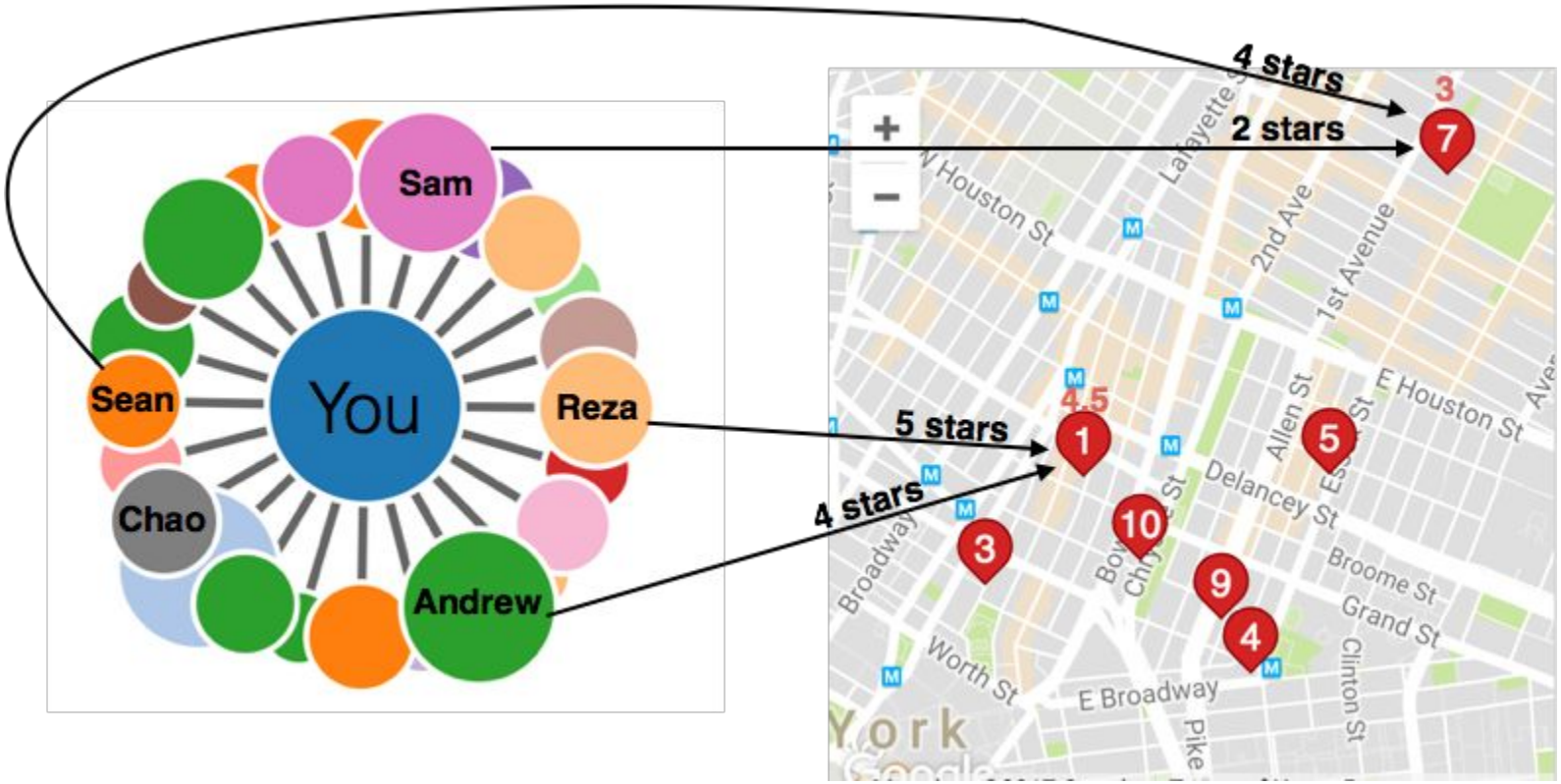
```
friends.show(50)
```

edgeDF

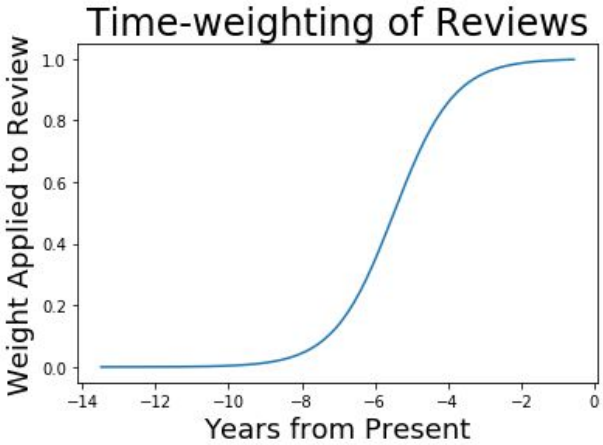
u0	u1
1003979	1024458
297574	1003979
970343	1003979
320810	1003979
188106	1003979
492231	1003979
352654	1003979



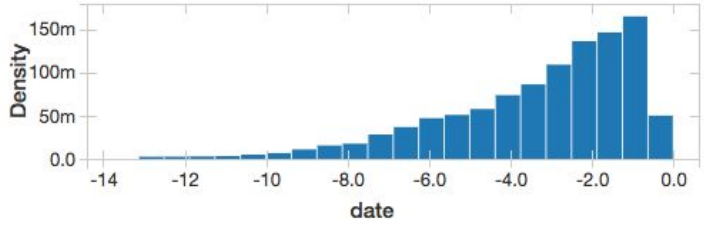
Yelp Social Network 2

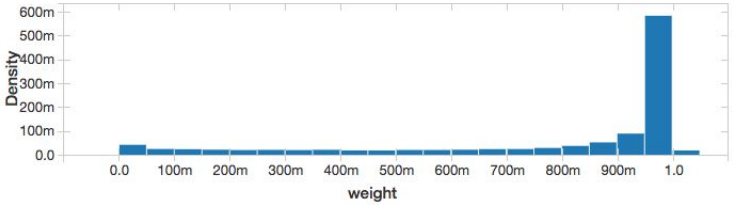


Review-level adjustments



Reviews binned by years from present



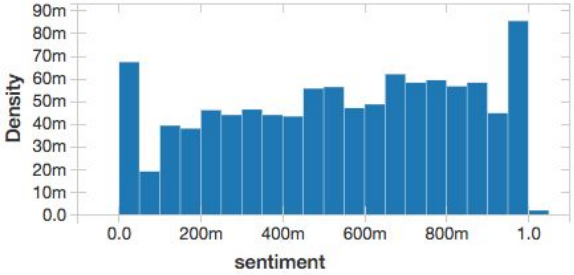


Newest 60% of reviews unaffected by time-weighting

How relevant is a 10 year old review?

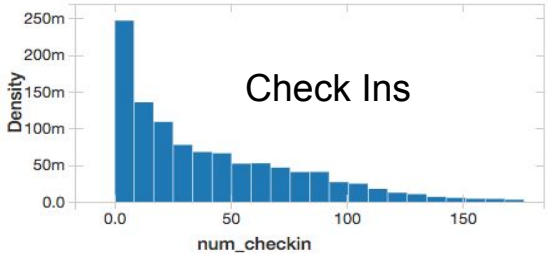
Business-level adjustments

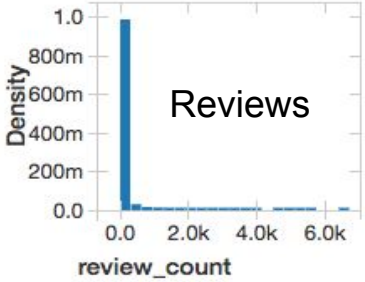
Multiplicative factor based on NLP sentiment



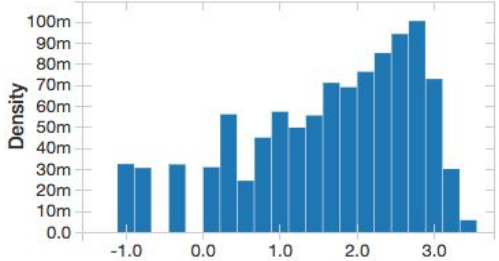
$$\xi = \left(\frac{3}{4} + \frac{\text{sentiment}}{4}\right) \cdot \sum_{\text{all reviews}} f(t, \text{rating})$$

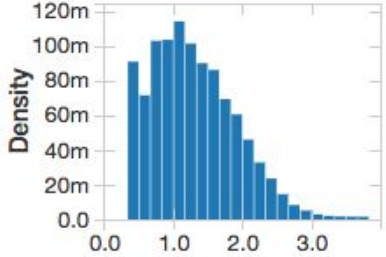
Log transformations for popularity measures





Log

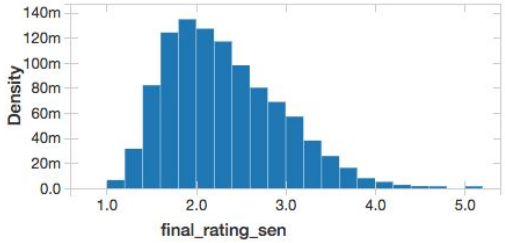


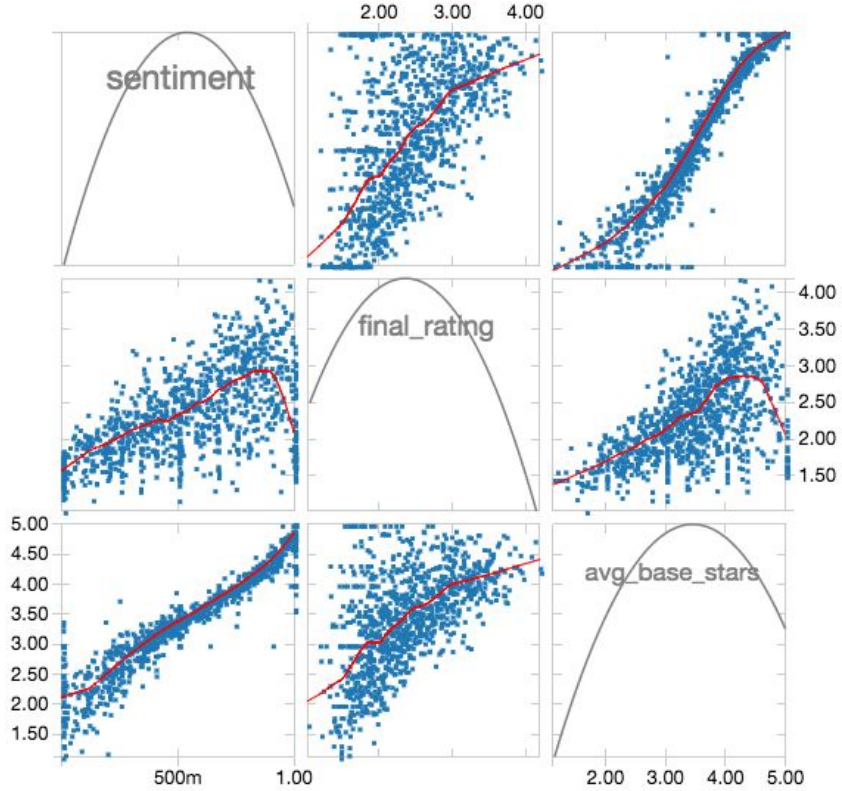


$$\text{final score} = \log_{10}\left(\text{review count}^{\xi} \cdot g(\text{check-in count})\right)$$

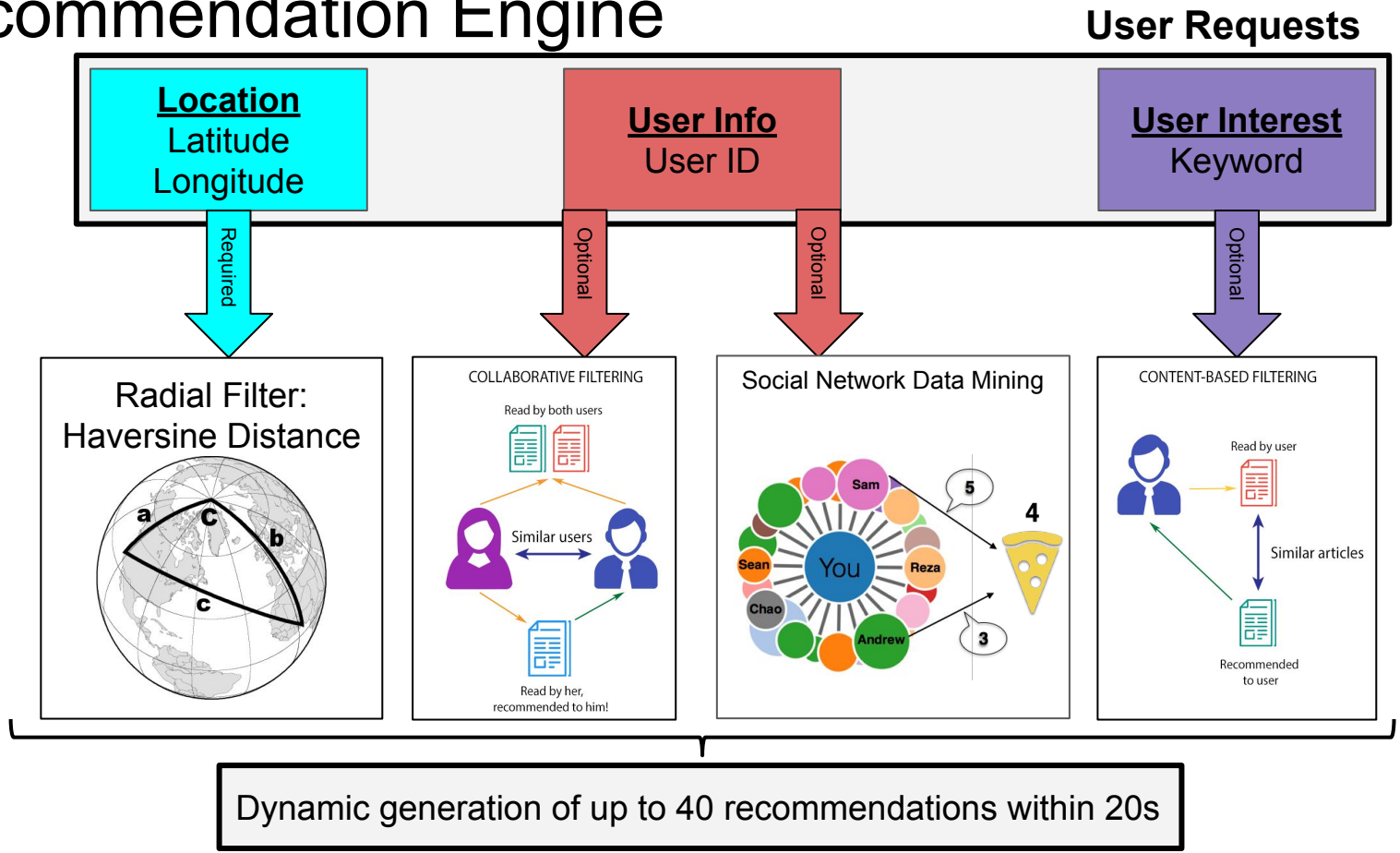
Final Location-only Ratings

Stars	Percentile
★★★★	99th (top 40)
★★★½	95th (top 300)
★★★	83rd
★★½	60th
★★	31st
★½	5th

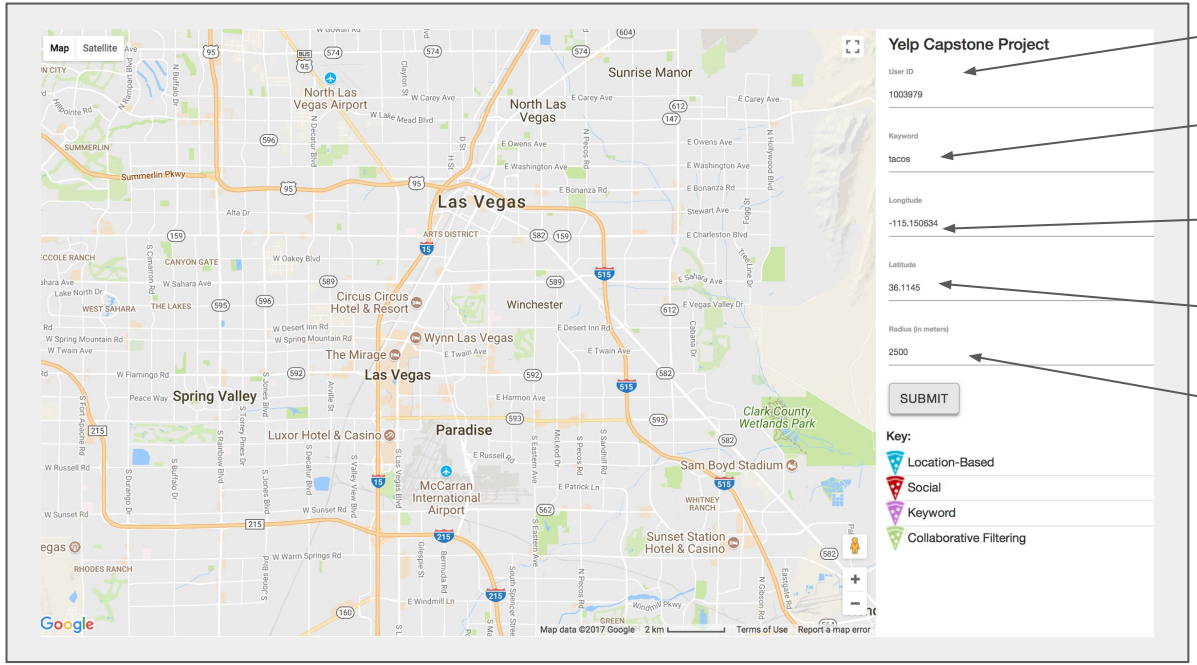




Recommendation Engine



Data Pipeline: 1. Flask



User ID

Keyword

Longitude

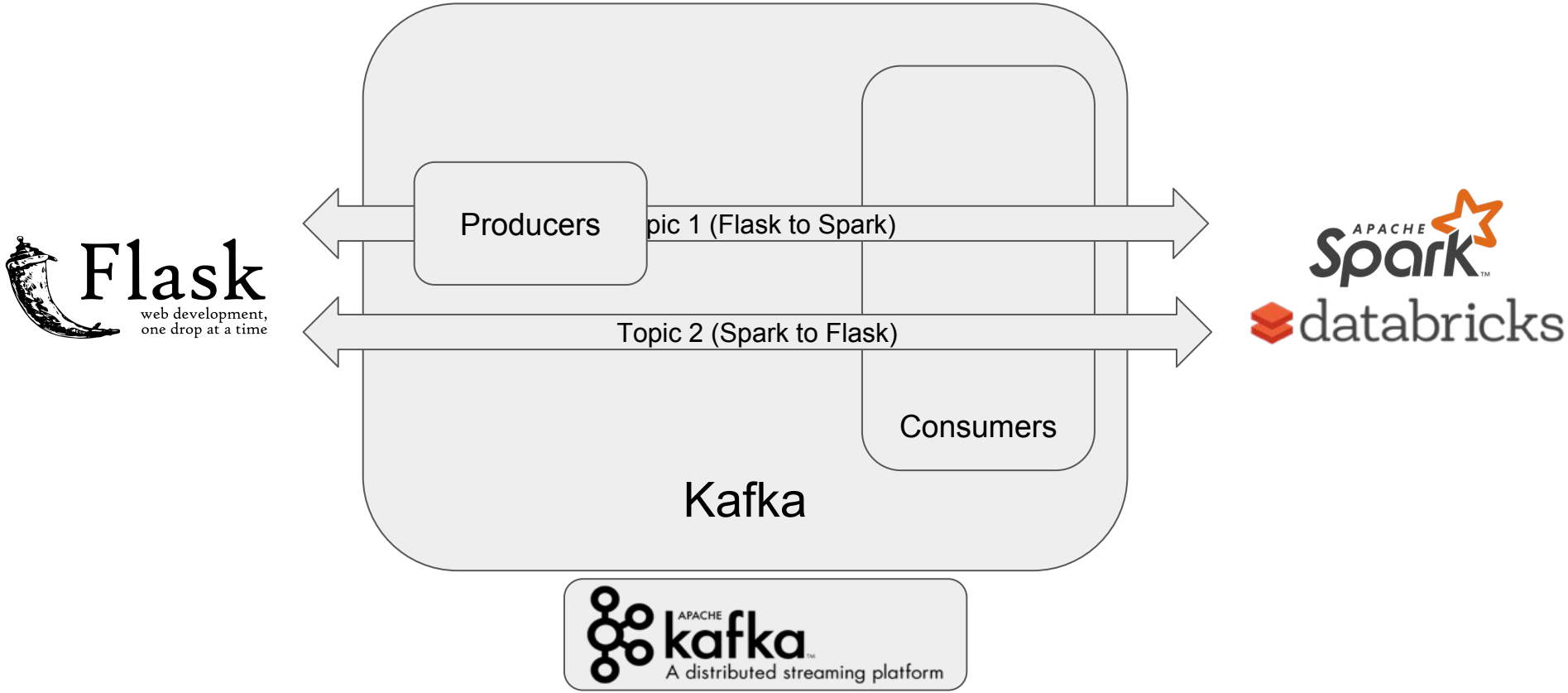
Latitude

Search Radius

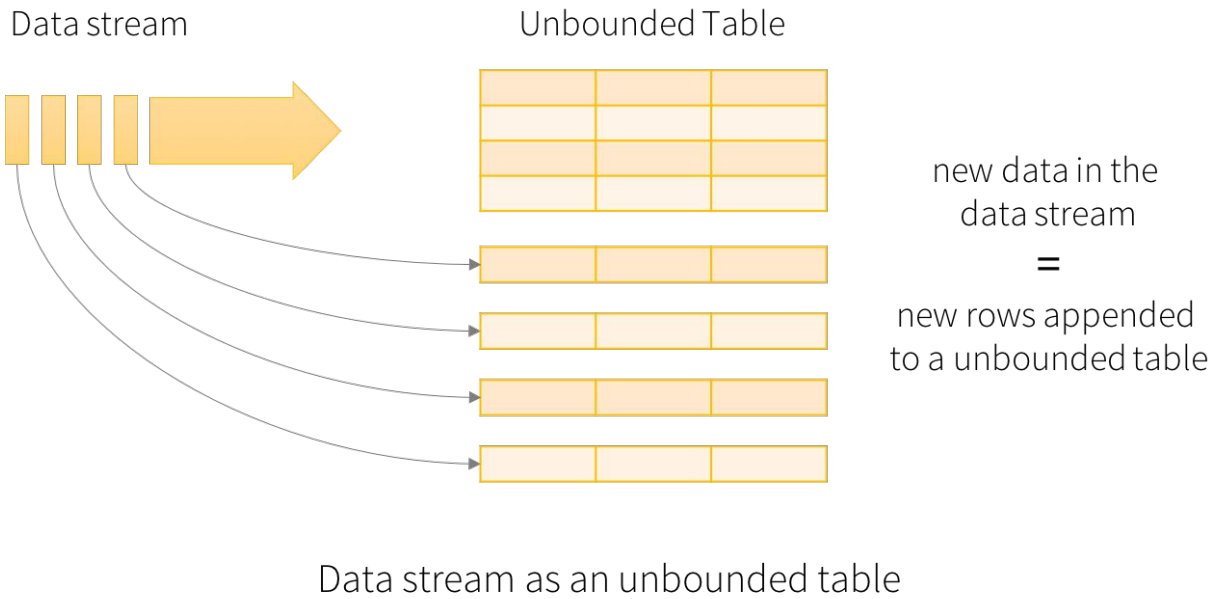
Flask
web development,
one drop at a time

So What’s Kafka?

Data Pipeline: 2. Kafka



Data Pipeline: 3. Spark



Spark's Structured Streaming

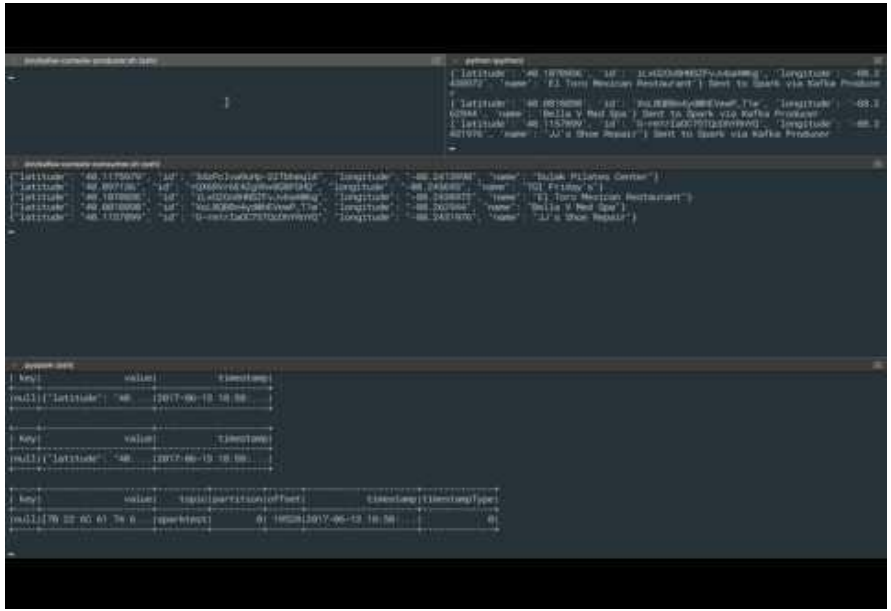
Data Pipeline: Overall

Flask
web development,
one drop at a time

APACHE kafka
A distributed streaming platform

APACHE Spark

databricks



Demo

Lessons Learned and Future Work

Lessons

- Recommendation Algorithms
- Dealing with Big Data
- Working in a Startup-like Environment

Future Work

- Front End:
- Back End:

Acknowledgement

Guidance and Discussion from

- Shu Yan
- Yvonne Lau
- Zeyu Zhang

Inspiration from

Chuan Sun and Aiko Liu

Thanks!