

# E6893 Big Data Analytics:

## *Eating Mate Recommendation*

Team Members: Zehao Song, Hao Xu, Yutong Tie



December 15, 2016

- ✓ Motivations
- ✓ Overview of the algorithm
- ✓ Technologies Used
- ✓ Dataset Overview
- ✓ Challenges / Struggles
- ✓ Findings
- ✓ Demos
- ✓ Conclusion
- ✓ Future Work

## ➤ Background

Yelp is an excellent application to help people find restaurants. People sharing their experiences during the meal on Yelp and Yelp makes recommendations for good restaurants based on people's rating history. However, Yelp cannot recommend people have the same eating habit to have meal together currently.

## ➤ Target

- ✓ Collect user data
- ✓ Collect review data
- ✓ Collect business data
- ✓ Make eating mate recommendations
- ✓ City taste

### Review of the Day



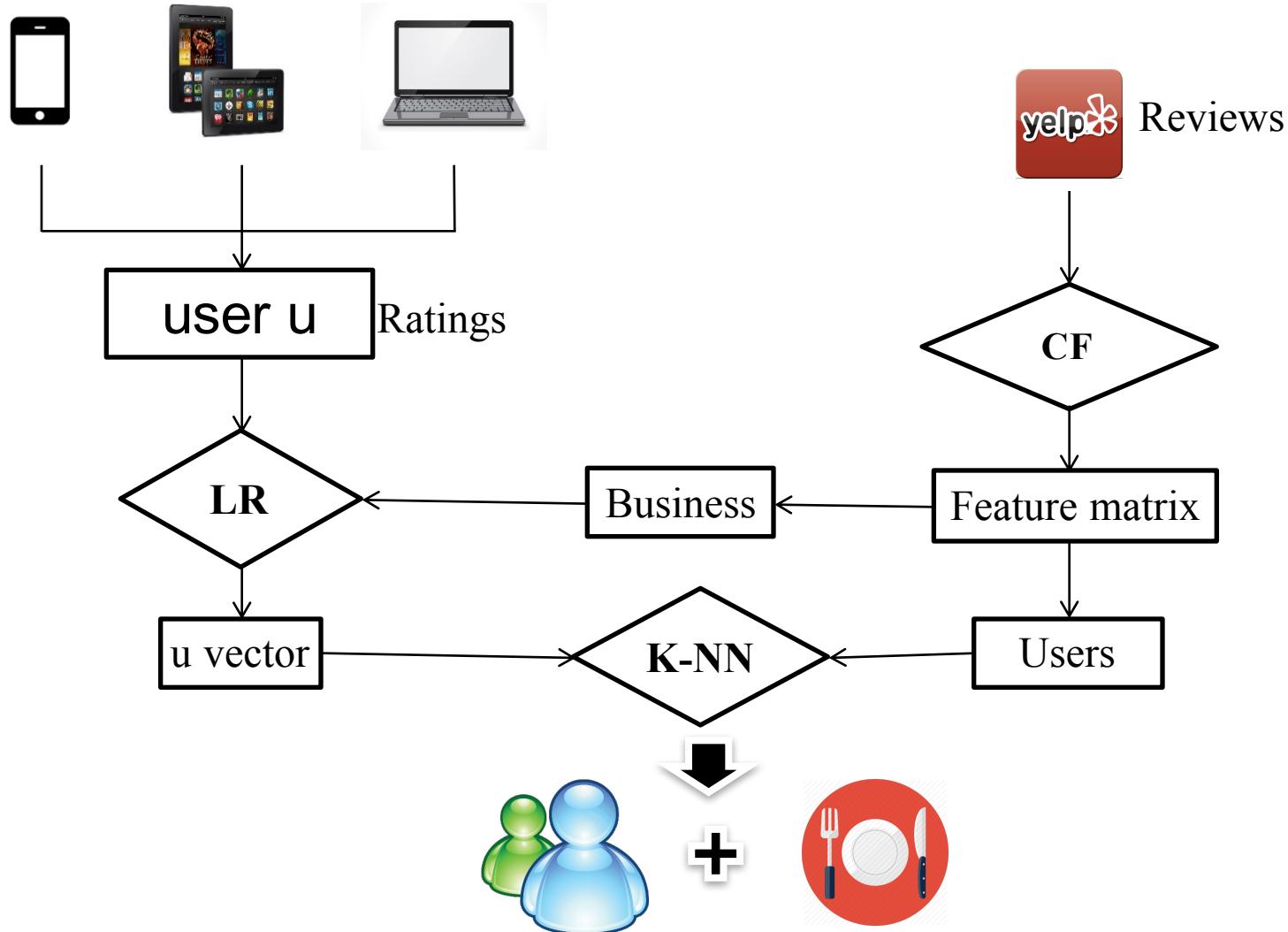
Rachel W. wrote a review for  
**Dinnertable**



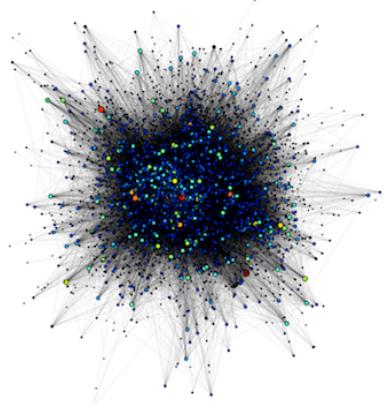
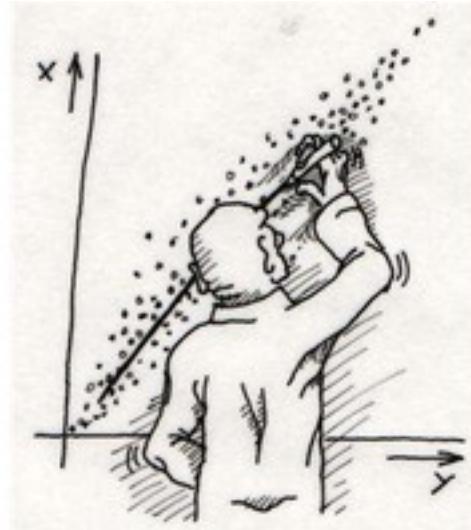
7/18/2016

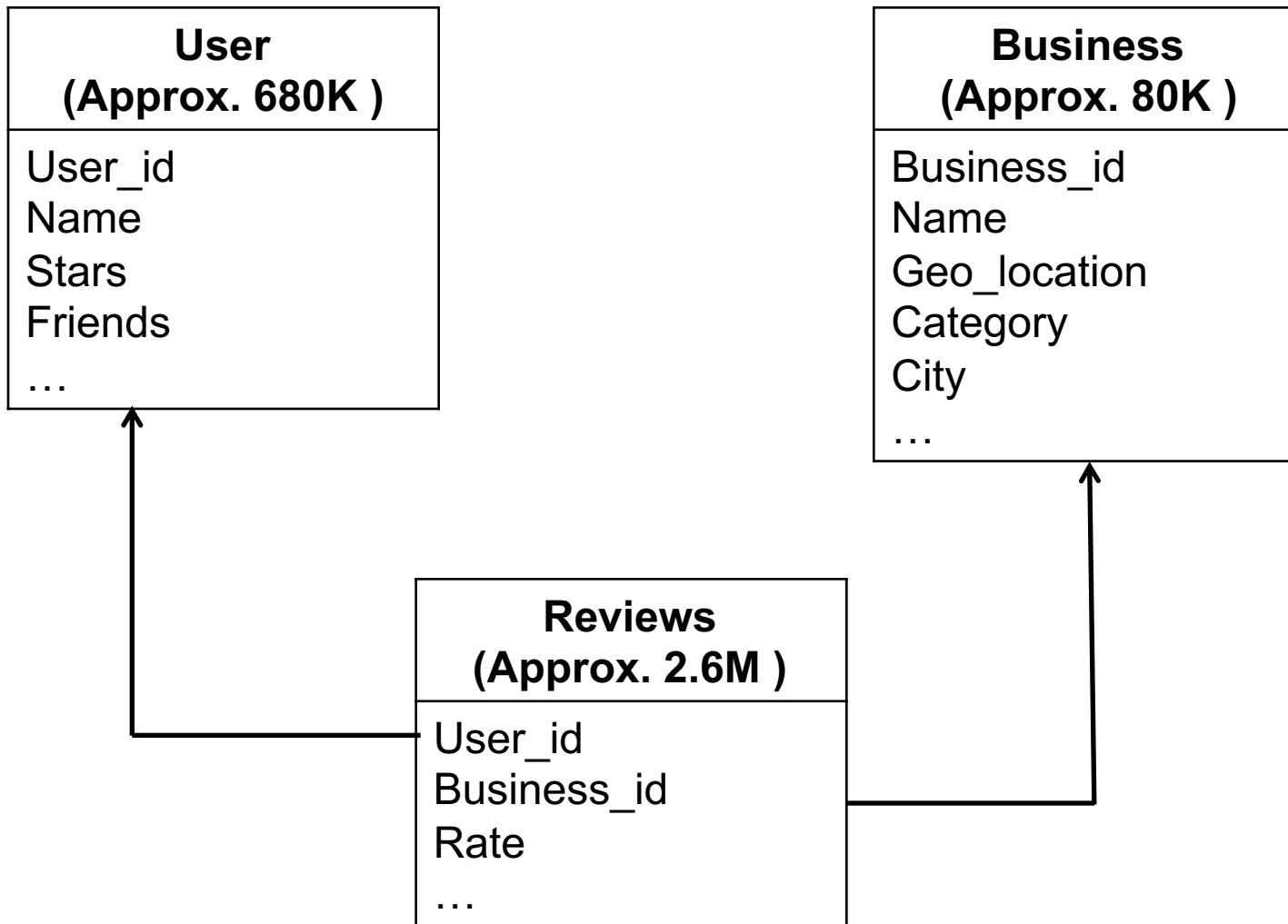
I'll just start off by saying this place is great! For anyone that doesn't know Dinnertable is like a reverse speakeasy of... [Read more](#)

# Eating mate recommendation system



1. Yelp Search API.
2. Scrapy.
3. Spark to analysis:
  1. Collaborative filtering
  2. K-NN
  3. Linear regression





1. It is hard to get review dataset with Yelp API:

Solution:

1. Use API to get Business ID
  2. Use Business ID to generate URL and use scrapy to get data.
2. Frequently send request to Yelp results in a block:

Solution:

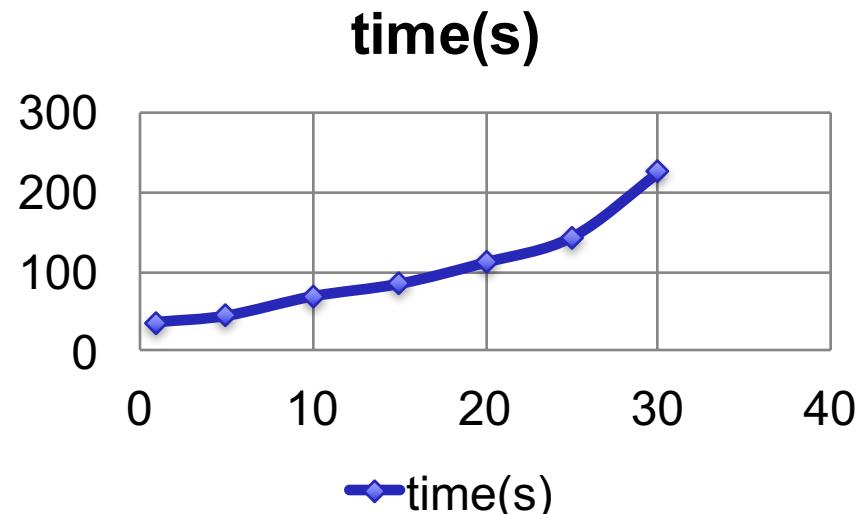
Frequently change IP address and combine data.

3. It is hard to evaluate recommendation result.

1. Parameter rank influences the performance of collaborative filtering:

Solution:

Hold-out validation.



Finally, rank 25 is selected taking both training error and time into consideration!

## 2. City tastes:



# pittsburgh



## Charlotte

# Example: Recommendation for Nader (User\_id=9)

## New User - Provide a series of ratings

```
# Only show the first 10 records
test_ratings.filter(lambda r: r[0] == 9).map(lambda r: (r[0], r[1], r[2])).collect()[:10]

[(9, 9213, 3.0),
 (9, 9487, 3.0),
 (9, 9513, 5.0),
 (9, 9659, 5.0),
 (9, 9754, 1.0),
 (9, 9811, 4.0),
 (9, 10252, 3.0),
 (9, 10345, 4.0),
 (9, 10376, 1.0),
 (9, 10641, 1.0)]
```

Business_name	Category	Rate
Red robin	American	3.0
Dino's Lounge	Bar	3.0
Luv It Frozen Custard	Ice cream	5.0
Sahara Korean Restaurant	Korean	5.0
Thai Grill	Thai food	4.0

## Recommend the Eating Mates of the User

```
evalEatingMateRecomd(9, test_ratings, trainModel)
```

```
417526 82% [71885, 71912, 6359]
549161 81% [71885, 6359, 78091]
  54383 79% [71885, 71912, 6359]
607208 79% [8882, 6359, 81313]
   2617 78% [71885, 71912, 6359]
480618 77% [8882, 71885, 71912]
566543 77% [78091, 71885, 71912]
611100 77% [71885, 71912, 6359]
530940 76% [71885, 6359, 71912]
673693 76% [71885, 71912, 6359]
Time Cost: 33.87s
MSE of User Vector: 1.17
```

User_name	Restaurant	Category
Will	EI Barrio	Bar
Will	Stage Bar	Bar
Will	Forum Cafe	American

- By selecting correct parameters on collaborative filtering, we are able to make prediction both correctly and efficiently.
- Current eating mate recommendation system is able to make recommendation based on user's rating history.
- Apart from making recommendation, we are able to analysis tastes for each city.

- Future Enhancements:
  1. Extend the system to more cities.
  2. Enhance the scalability of the system.
  3. Have a dashboard to display some interesting features, like city taste.
  4. Collect feedbacks from users about recommendation to make modifications on algorithms.

A large, colorful word cloud centered around the words "thank you" in various languages. The word "thank" is in red, "you" is in white, and "thank you" together is in red. The surrounding words are in different colors and fonts, representing numerous languages from around the world. Some examples include "danke" in German, "спасибо" in Russian, "merci" in French, "gracias" in Spanish, "mochchakkeram" in Korean, and "xie xie" in Chinese.