# The Yelp Machine

NYC Data Science Academy | Capstone Project | Team PC1

Aiko Liu | Amy Chen | David Steinmetz | Greg Domingo

# The team

## Aiko Liu

With quantitative training in math/physics, focus on the application of machine learning techniques to finance, big data and beyond

## Amy Chen

Devoted to using data visualization and machine learning techniques for social and business innovations

## David Steinmetz

Passionate about creating value by distilling data into actionable information, particularly through visualization

## Greg Domingo

Keen interest in innovation with Data Science as one of the leading edges of innovation space

# Agenda

Overview and Context

Explanation of the App

Next Steps

# Agenda

Overview and Context

Explanation of the App

Next Steps

# You want to go out to eat with a friend but sifting through restaurant listings is frustrating and difficult

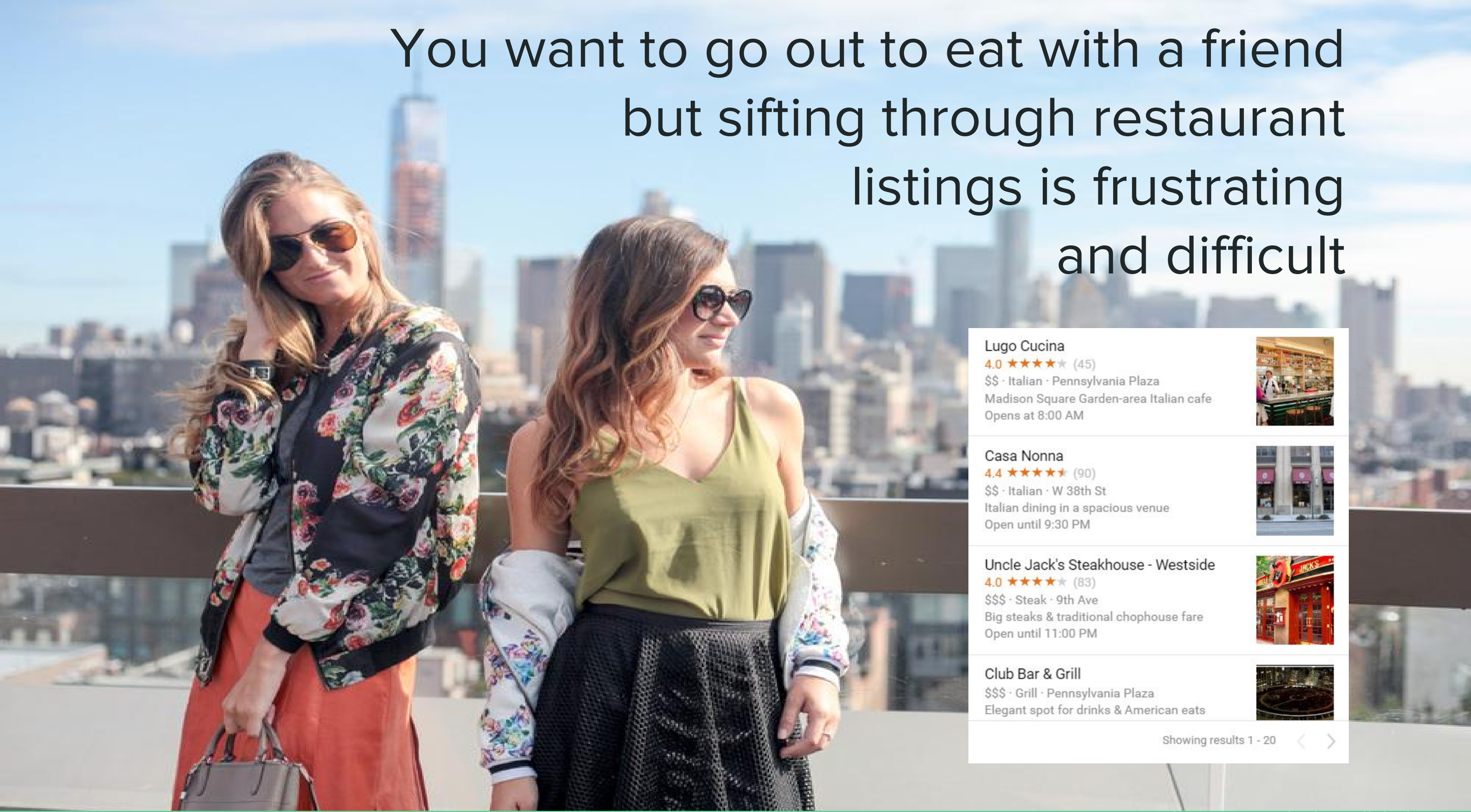
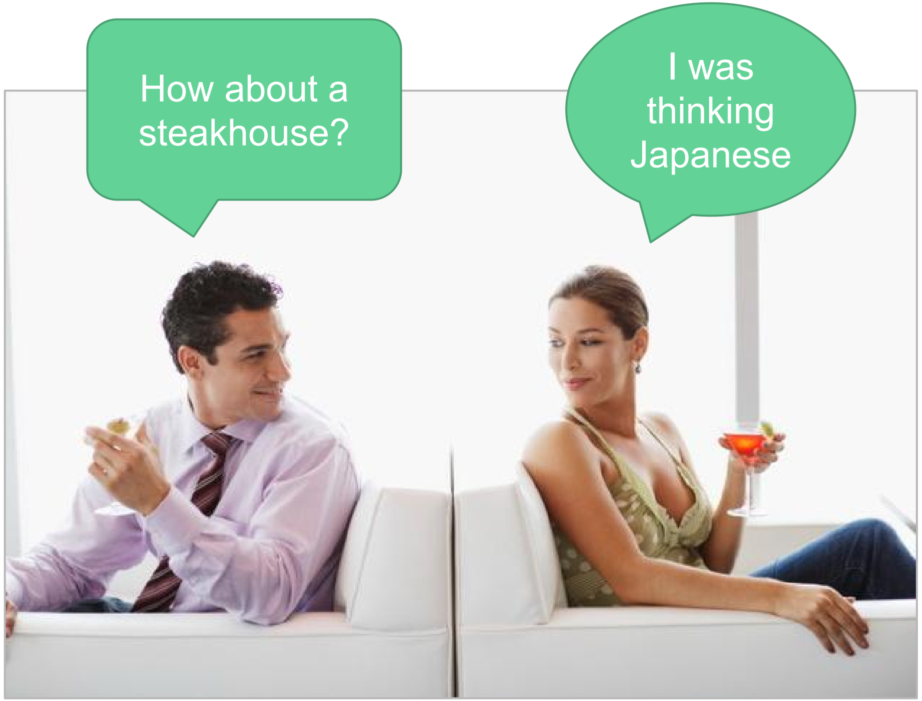
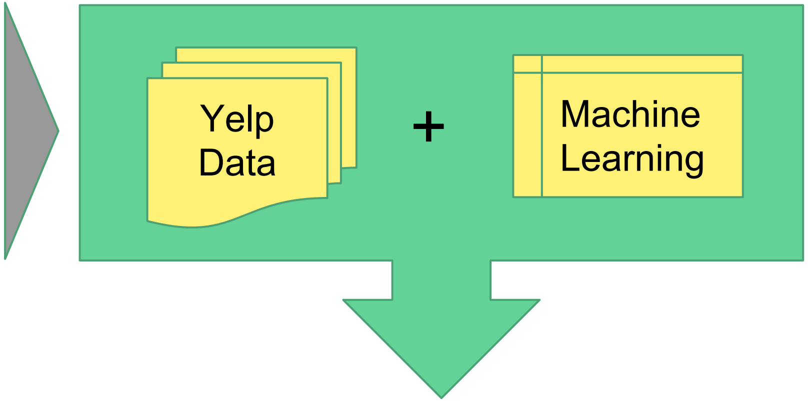
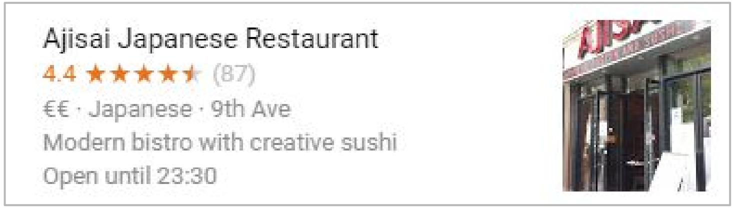


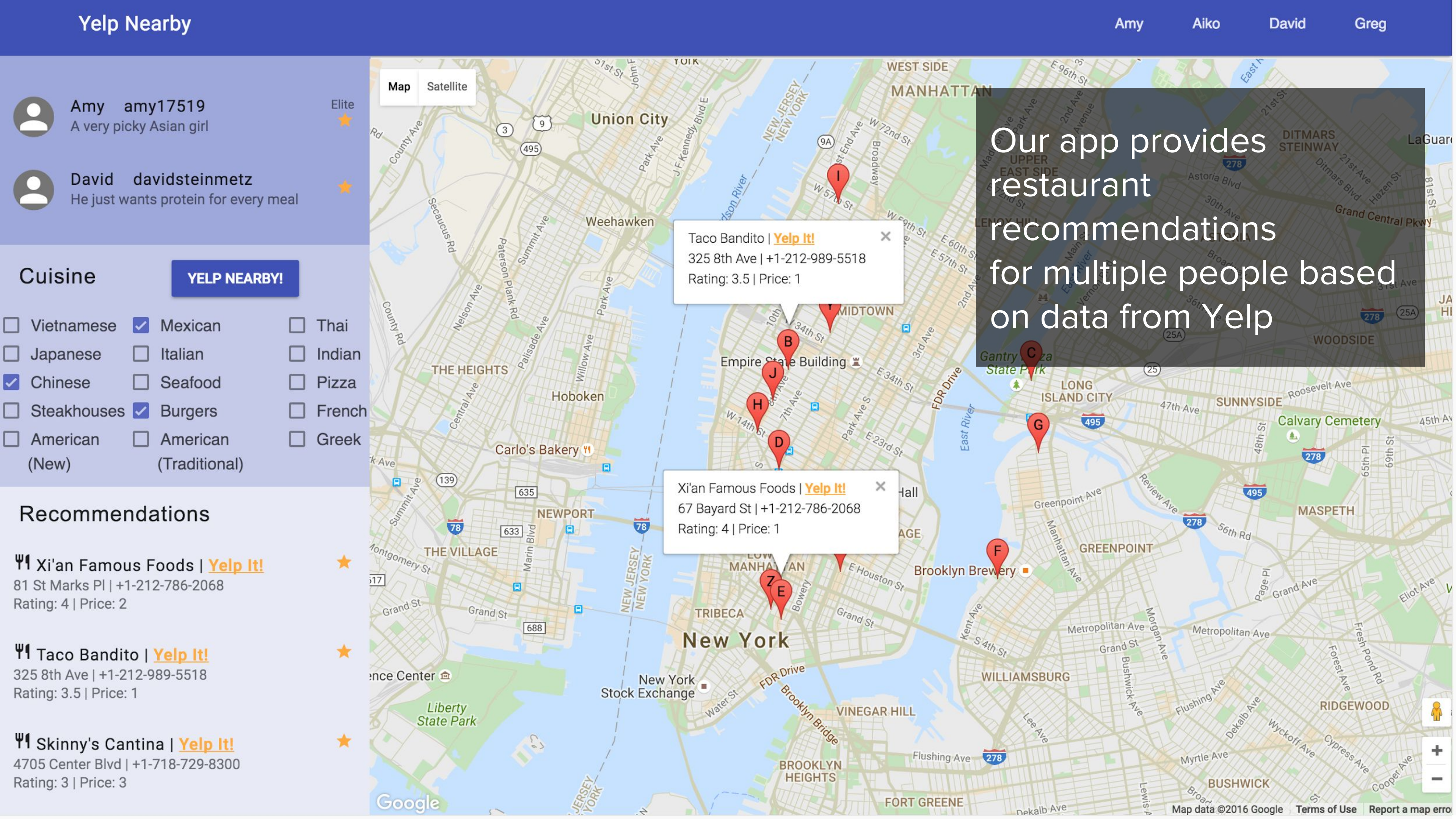

Image: www.locksandlabels.com

# Could Yelp data be used in conjunction with machine learning to find a restaurant which will suit the tastes of two people?




Yelp Nearby
Amy
Aiko
David
Greg
Amy amy17519
A very picky Asian girl
Elite
David davidsteinmetz
He just wants protein for every meal
Cuisine
YELP NEARBY!
Vietnamese
Mexican
Thai
Japanese
Italian
Indian
Chinese
Seafood
Pizza
Steakhouses
Burgers
French
American (New)
American (Traditional)
Greek
Recommendations
Xi'an Famous Foods | Yelp It!
81 St Marks Pl | +1-212-786-2068
Rating: 4 | Price: 2
Taco Bandito | Yelp It!
325 8th Ave | +1-212-989-5518
Rating: 3.5 | Price: 1
Skinny's Cantina | Yelp It!
4705 Center Blvd | +1-718-729-8300
Rating: 3 | Price: 3
Map
Satellite
Taco Bandito | Yelp It!
325 8th Ave | +1-212-989-5518
Rating: 3.5 | Price: 1
Xi'an Famous Foods | Yelp It!
67 Bayard St | +1-212-786-2068
Rating: 4 | Price: 1
Our app provides restaurant recommendations for multiple people based on data from Yelp
New York
Union City
Hoboken
Weehawken
Empire State Building
Map data ©2016 Google
Terms of Use

# Agenda

Overview and Context

Explanation of the App

Next Steps

*Explanation of the App*

# Our App marries a Flask front end with a Python back end to provide recommendations

## Front end

**Flask**
Python
Microframework

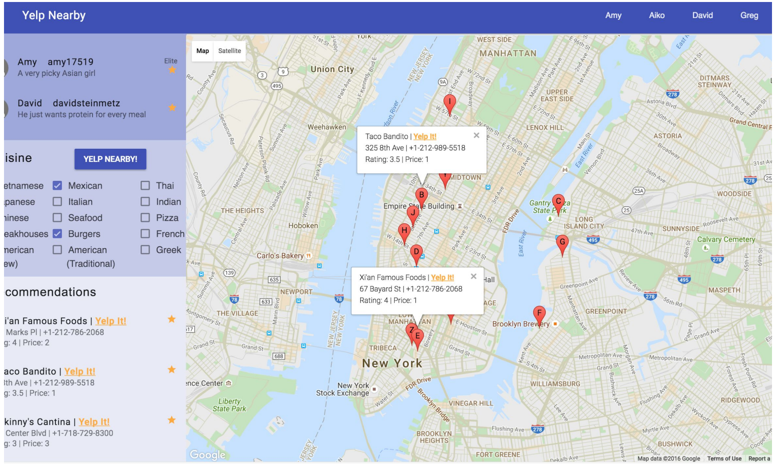
**Jinja2**
Templating

+

**HTML / JavaScript**
Programming
Languages

## Yelp Nearby

A multiuser restaurant recommendation engine




## Back end

**Python**
Classes,
Functions,
Modules

**Ext**

**Yelp**
API

+

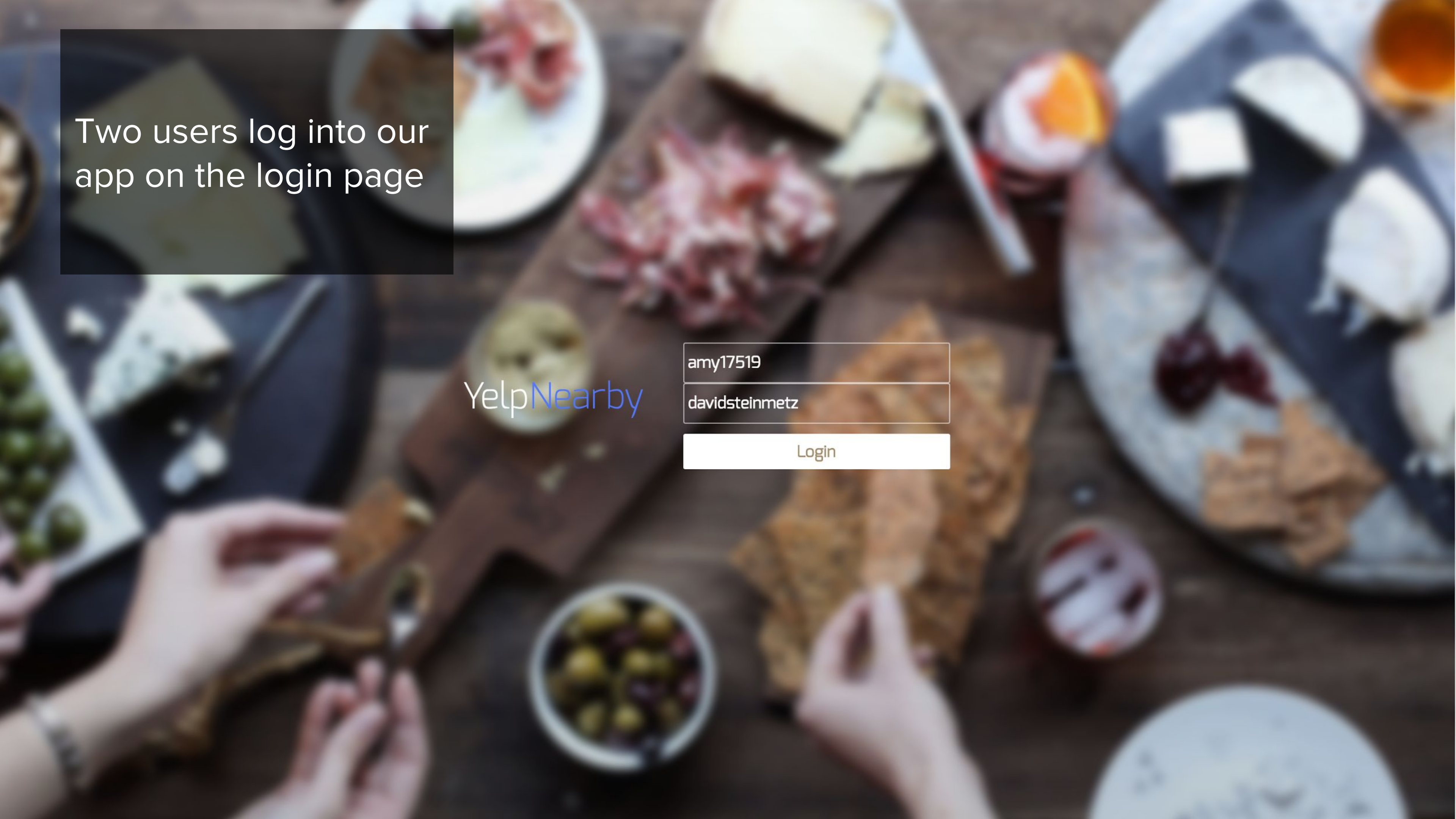
**Internal**

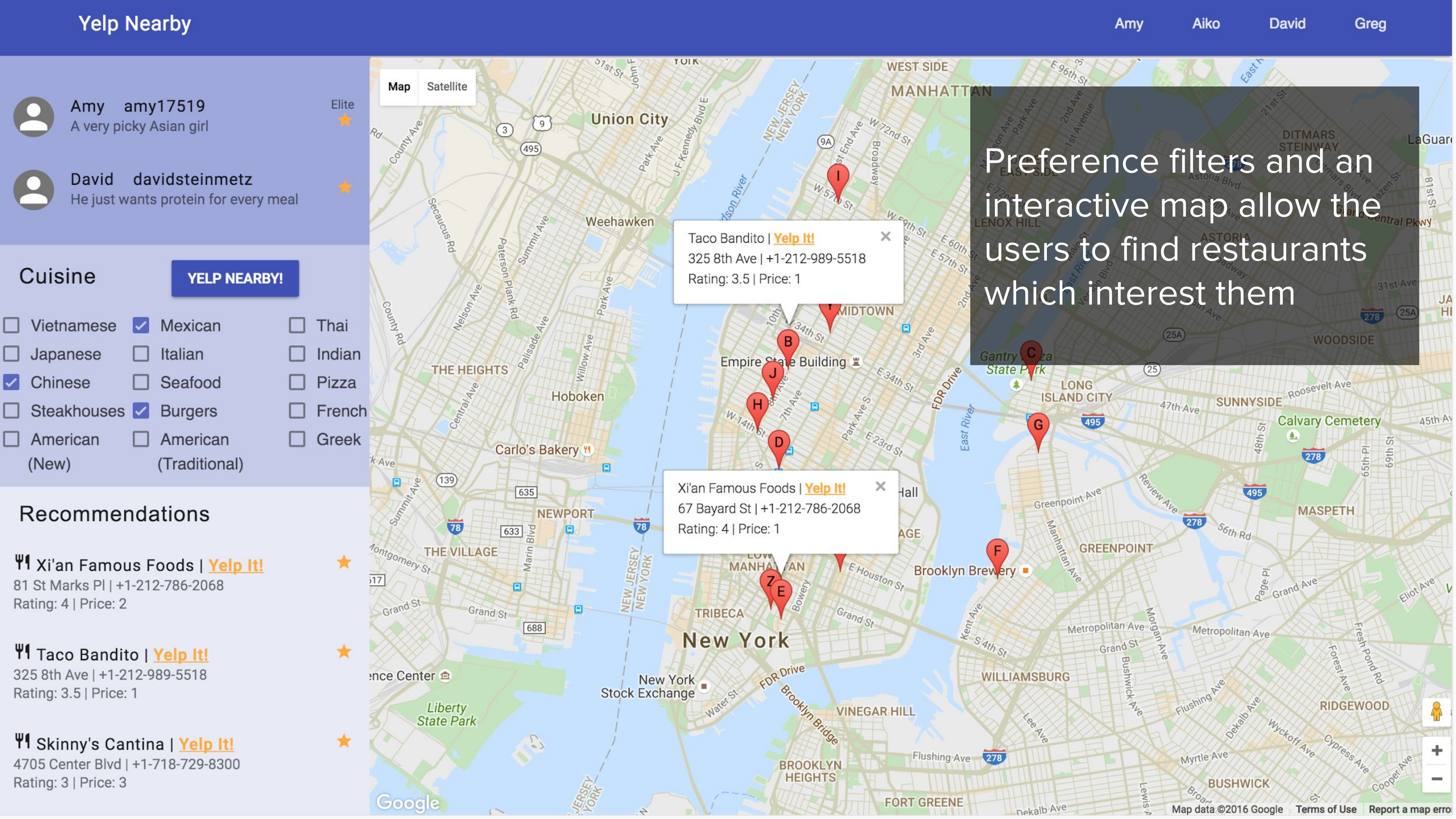
**GraphLab**
Machine Learning

+

**Homemade**
Class/Func/Mod

Two users log into our app on the login page





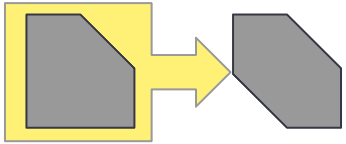

Preference filters and an interactive map allow the users to find restaurants which interest them

# The recommendation system works in a pipeline of three processes

| 1 Train | 2 Cluster | 3 Classify |
| --- | --- | --- |
| • Data about users, business, reviews from online Yelp Challenge<br>• Collaborative Filtering recommends restaurants for specific users | • Clusters needed to extend model to new locales<br>• Density-based scanning used to create clusters<br>• Restaurants are clustered based on selected features | • Locally available restaurants are classified into the clusters of the predicted recommendations |

# A collaborative filtering model was chosen because it incorporates information from users who make similar reviews

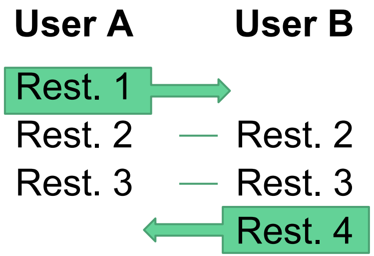
## Content-based systems



Predicts similar items

| Advantages | Disadvantages |
|---|---|
| 1. Uses the items' content to predict the user's interest<br>2. Recommendation quality improves as the review/item content data cumulates | 1. Impossible to predict the totally distinct types of items the particular user has never expressed interest in<br>2. 2. Limited by the collected items' info in making recommendation (New Item?) |

## Collaborative filtering

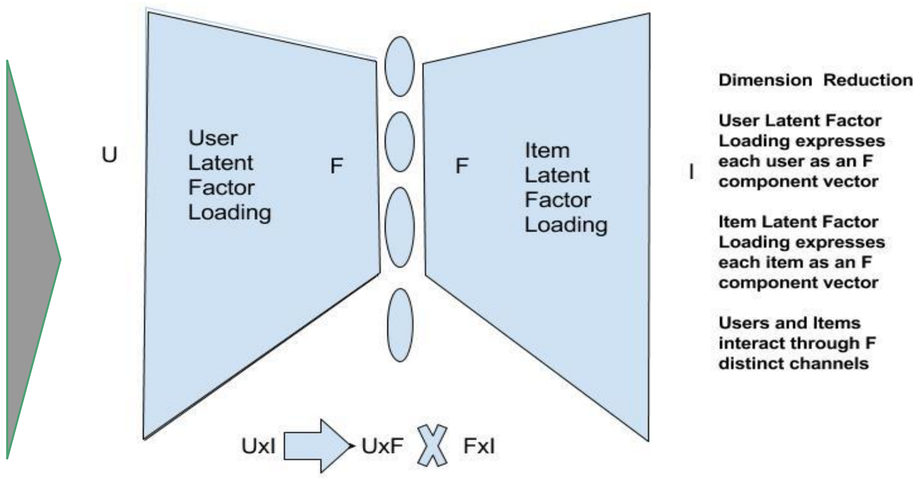



Predicts items from user preferences and from similar users

| Advantages | Disadvantages |
|---|---|
| 1. Predict items through similar user patterns, even if the particular user has a short review history<br>2. Works without item attributes<br>3. 'Outside the Box' recommendation | 1. Cold Start for the new users<br>2. Sparse Ratings on the same item<br>3. Recommendations are difficult for users with distinct tastes; these users are called black sheep or gray sheep. |

# Latent Matrix Factorization is the key component of collaborative filtering

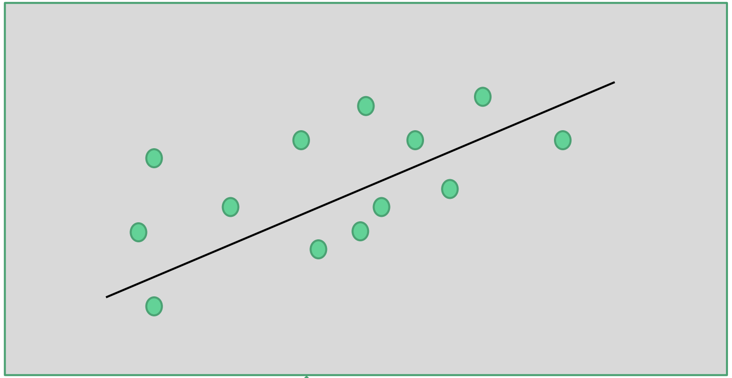
| | Rest. 1 | Rest. 2 | Rest. 3 | Rest. 4 |
|---|---|---|---|---|
| User 1 | 5 | | 2 | |
| User 2 | | 3 | 4 | 1 |
| User 3 | 1 | | | 4 |

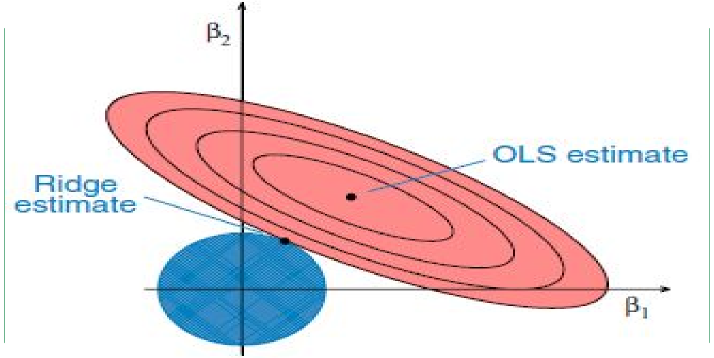



Numbers in the table are the rating the user gave the restaurant on a scale of 1-5

The matrix to the left is factorized

# Latent Matrix Factorization adds to two well-known machine learning techniques: linear regression & L2 regularization






Linear regression | Latent Matrix Factorization | Ridge regression

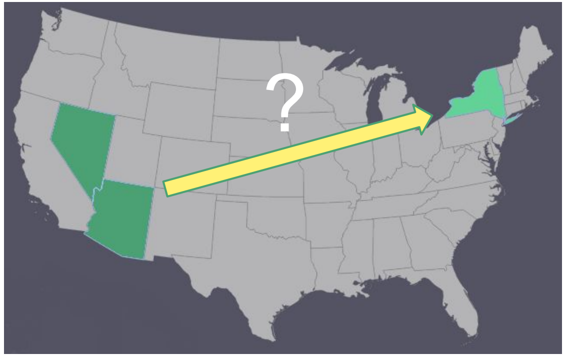
1. Matrix Factorization, the core of CF, can work without side inputs from the users, items, capturing the user-item interaction through factorizing the sparse user-item matrix
2. The linear regression upon the side information reduces the model estimation residuals
3. The L2 regularization, known as Ridge regression in the context of MLR, controls the stability of the model fit and prevents over-fit
4. Three parts are combined into the single equation system (graphlab)

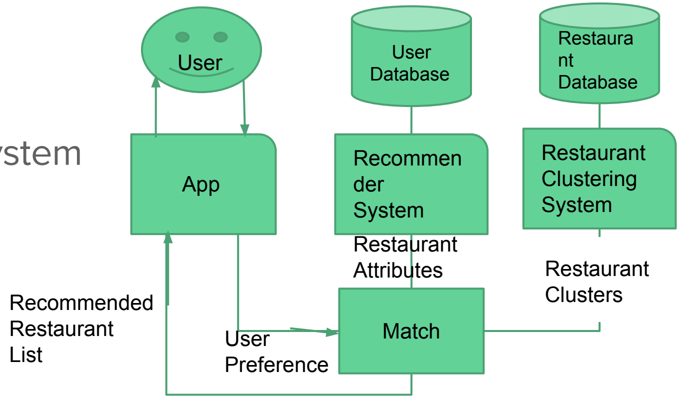
# Additional features can be included as side information in Latent Factorization to train the model

| Feature Name | Feature Equation | Why it’s included |
|---|---|---|
| User_EliteYears | 1 * years_elite | Elite users have outsized influence on ratings |
| User_AvgRating | mean(rating) | Different users have different rating standards |
| User_Num_Review | log(u_num_reviews+1) | The indicator of the user’s engagement on yelp |
| User_Location | city/state of the reviews | The reviews from the same location may be similar |
| Rest_AvgRating | mean(user rating) | The reviews’ consensus on the restaurant quality |
| Rest_Num_Review | log(r_num_reviews+1) | The attention the business get from the reviewers |
| Rest_Aggr_EliteYear | Sum of the | The attention the business receives from the leaders among the reviewers |
| Rest_Location | city/state of the Rest. | The location of the restaurants are highly correlated with the residence of the reviewers |

# Making restaurant recommendations outside the limited region in the dataset posed a problem

- Recommender system only maps to restaurants in the original dataset
    - Original dataset does not include major cities like New York and San Francisco
- So how do we recommend restaurants outside the areas in the dataset or in areas with very few reviews
- Solution: Cluster Analysis
- Solution Concept:
    - Get attributes of restaurants selected from recommender system
    - Match those attributes with results of cluster analysis to determine cluster assignments
    - Get restaurants in the area of current user which fall in the selected cluster

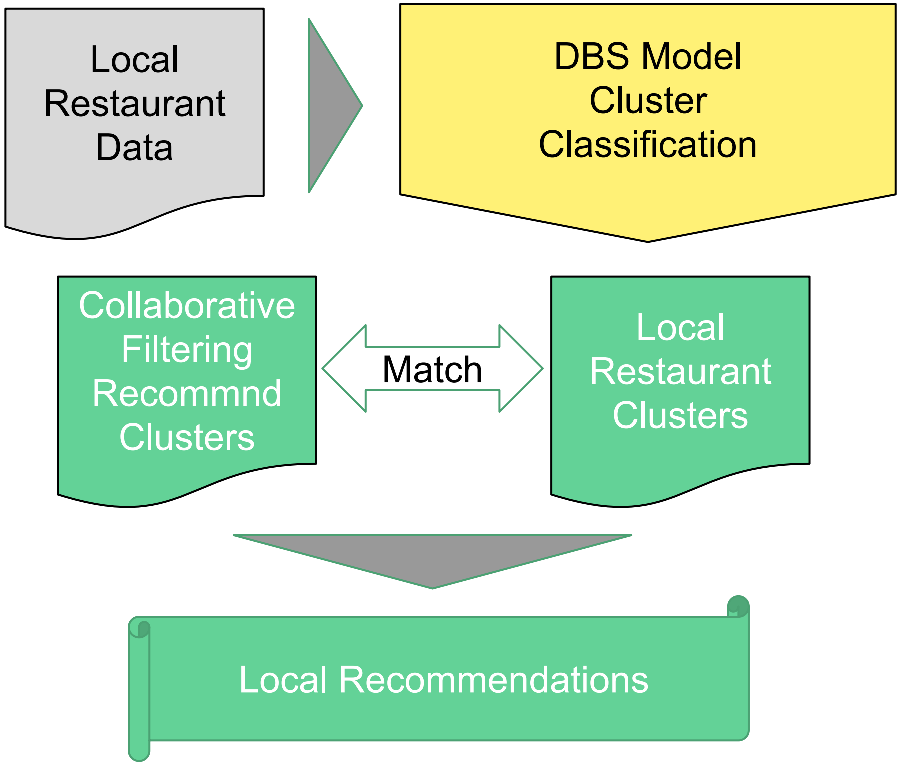


# Density-based scanning was chosen to cluster all restaurants in the data set

| **Algorithm** | **Advantages** | **Disadvantages** |
|---|---|---|
| K-Means | • K-means works well when the shape of clusters are hyper-spherical<br>• Computationally efficient | • May give different results every time it is run<br>• Requires prior knowledge of number of clusters |
| Hierarchical Clustering | • Gives recommended clusters<br>• Repeatable results | • Time complexity is quadratic |
| **Density-based scanning** | • **Can handle clusters of different shapes and sizes**<br>• **Gives recommended clusters**<br>• **Computationally more efficient that hierarchical cluster method** | • **May have problem handling high dimensional data**<br>• **May have problem dealing with data that has widely varying densities** |

# Classifying locally available restaurants based on the DBS model solved the problem of a limited data set




| **Feature used** | **Description** |
|---|---|
| Ratings | Average Rating Of Restaurant Based on User Reviews |
| Price Range | Price Range For Restaurant |
| Review Counts | Number Of Reviews Of A Restaurant ( transformed using log function) |

# Agenda

Overview and Context

Explanation of the App

Next Steps

# The functionality of the app can be extended

- Extract information from the restaurant reviews using an NLP technique called Latent Dirichlet Allocation
  - This data can be included in the clustering model to improve distinction between clusters
- Use app users' reviews to improve recommendations
- Include new users not existing in the data set
- Extend to larger groups of users

Thank you
for your attention

# GraphLab Recommender Model

$$\text{score}(i,j) = \mu + w_i + w_j + \mathbf{a}^T\mathbf{x}_i + \mathbf{b}^T\mathbf{y}_j + \mathbf{U}_i^T\mathbf{V}_j,$$

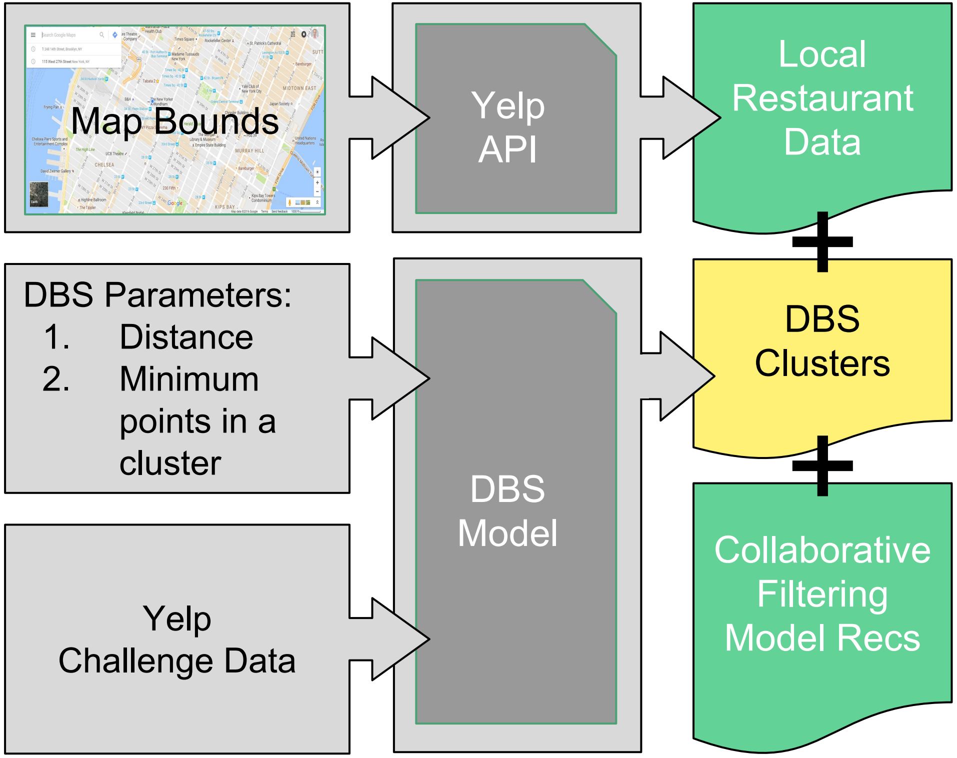
$$Objective = \min_{\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{V},\mathbf{U}} \frac{1}{|\mathcal{D}|} \sum_{(i,j,r_{ij})\in\mathcal{D}} \mathcal{L}(\text{score}(i,j), r_{ij})$$

$$+\lambda_1(\|\mathbf{w}\|_2^2 + \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2) + \lambda_2\left(\|\mathbf{U}\|_2^2 + \|\mathbf{V}\|_2^2\right)$$

$\mathcal{L}$ = (Squared Error) Loss Function,

$r_{ij}$ = rating of user $i$ to item $j$.

# Classifying locally available restaurants based on the DBS model solved the problem of a limited data set




| Feature used | Description |
|---|---|
| Ratings | Average Rating Of Restaurant Based on User Reviews |
| Price Range | Price Range For Restaurant |
| Review Counts | Number Of Reviews Of A Restaurant ( transformed using log function) |