# RECOMMEND RESTAURANTS FOR YELP USERS

DAT-NYC-25

TIM LIN

# PURPOSE

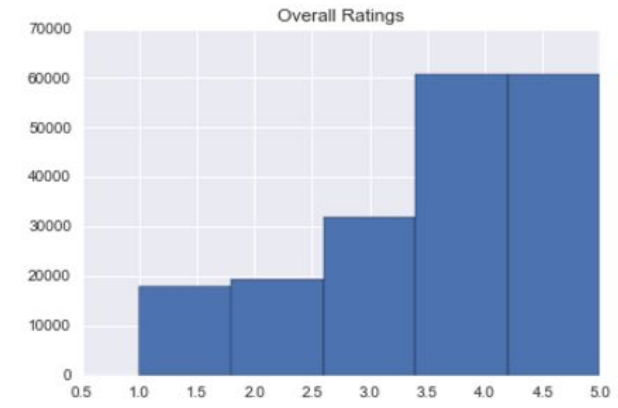
**Create a recommendation system based on user's review text and star ratings for restaurants in Las Vegas with the data provided by Yelp Data Challenge.**

**Can apply to millions of online products.**

# DATASET

**Yelp Challenge Dataset**

- 60K business, 1M reviews, 366K users across 10 cities.
- Only focus on food service business in Las Vegas, NV.
- Data:
  - 1.7K business with 2.5K unique IDs
  - 7.8k users with 190K reviews
  - Star rating from 1 to 5



# RECOMMENDATION SYSTEM

**Wiki Definition: Engines of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item.**

**Collaborative filtering:**

- Make recommendation based user preferences.

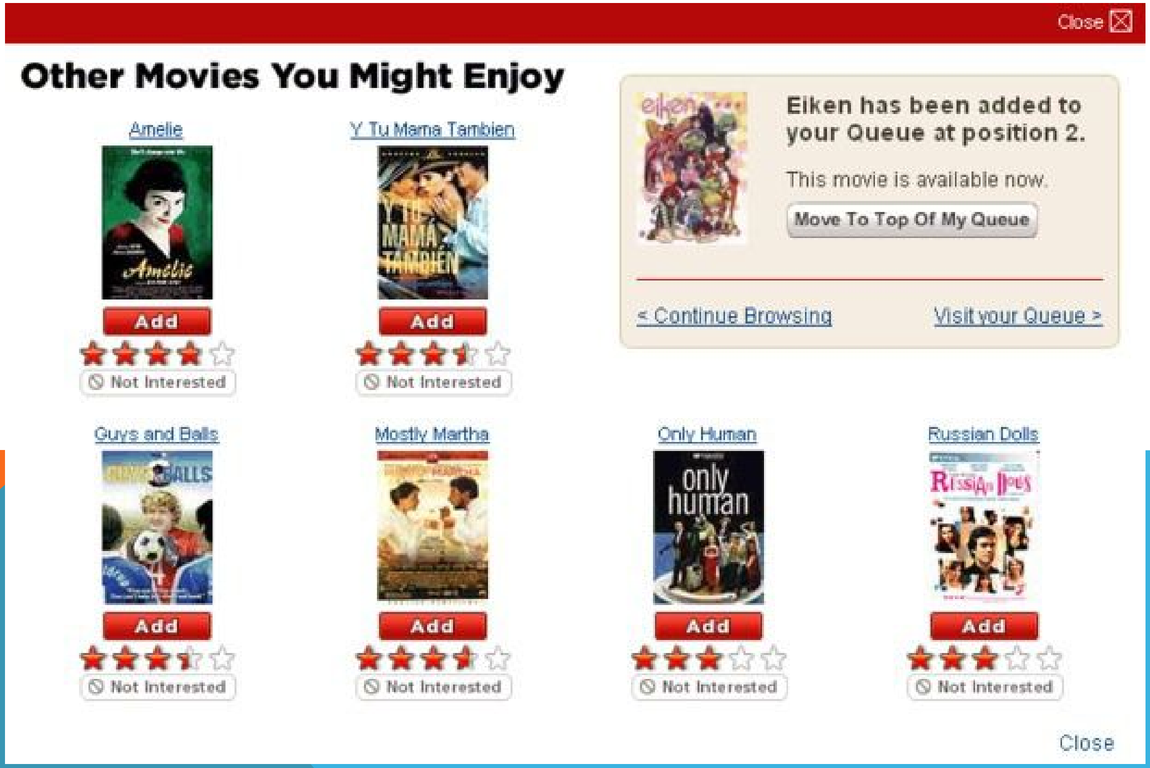
**Content-based filtering:**

- Recommend items based on items characteristics.

# COLLABORATIVE FILTERING (CF)
## "BECAUSE YOU LIKED THIS, WE THINK YOU'D ALSO LIKE THIS"

**Basic Assumption**

- User give ratings to an items
- Clients who had similar tastes in the past, will have similar tastes in the future.



# APPROACH

**Use the preference of crowd to recommend items**

**Methods:**

- Item-based
- Model-based

**Input:**

- Matrix of ratings given by users

**Output:**

- Top-n most similar items
- Prediction indicating likeness of a certain item

# ITEM-BASED CF

**Idea:**

- use similarity between items to make prediction.

**Approach:**

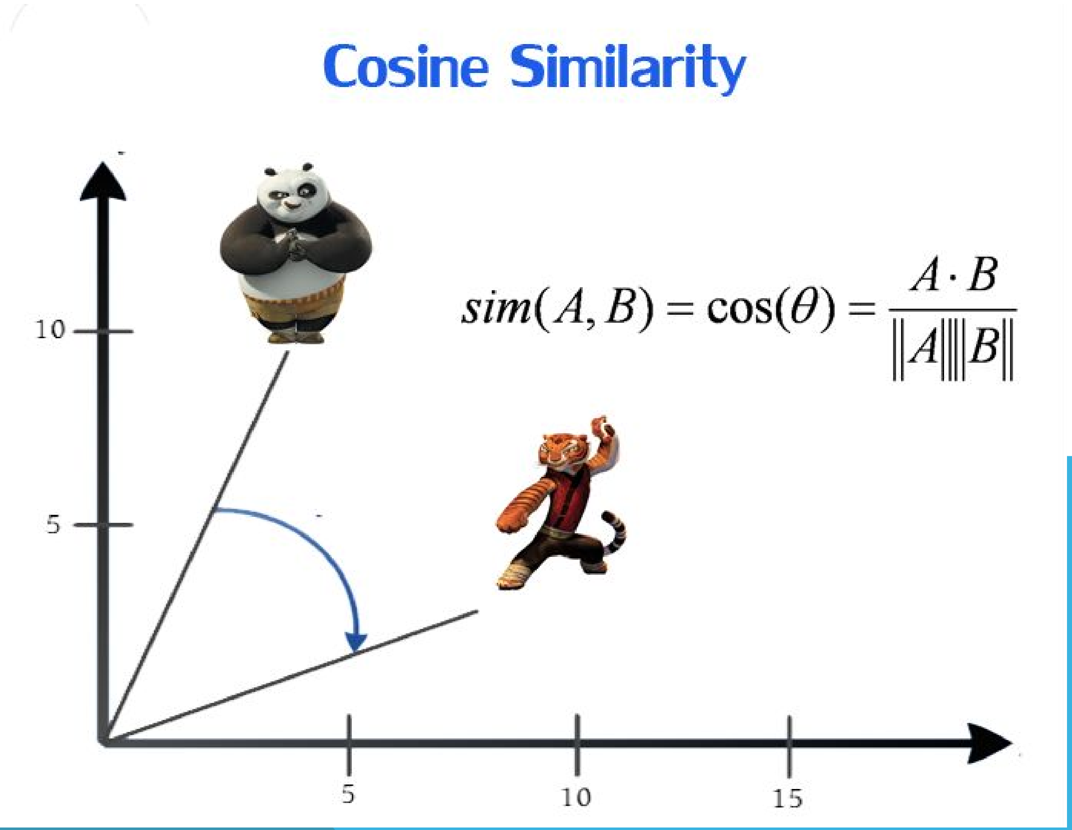
- Look for rating similar item 5.
- And use ratings for these item to predict the missing value.

| | Item1 | Item2 | Item3 | Item4 | Item5 |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

# MEASURE OF SIMILARITY

**Cosine Similarity**

- Items are put into vector in n-dimensional space
- Similarity is calculated based on the angle between the vectors



# MODEL-BASED CF

**Make prediction based on learned model (singular value decomposition, SVD)**

**Idea:**

- Reduce the dimension in data for faster prediction generation
  - SVD captures important features and place weights in data

**Pros:**

- Constant time to make recommendation.
- Filter out noise
- Detect nontrivial correlations in the data

**Cons:**

- Lost of information since original data is used.

# SINGULAR VALUE DECOMPOSITION
## (GOLUB AND KAHAN 1965)

**Any Matrix *M* can be decomposed into a product of three matrices:**

$$M = U \times \Sigma \times V^T$$

- *U* and *V* are left and right singular vectors which retain the most import features/ latent variables.

**Application:**

- both user and items are represented in term of these latent variables.
- Item vector characteristic of given features and user vector represent the preferences for a given feature.
- Rating are the sum of user vector and item vector by taking their dot products.

# USER:

| | |
|---|---|
| Swiss Cafe Restaurant | 4 |
| Spago | 4 |
| Samosa Factory | 4 |
| India Oven | 4 |
| Ichiza | 4 |
| Go Raw Cafe | 4 |
| Chianti Cafe | 4 |
| Baladie Café | 4 |
| Bonjour Bakery & Deli | 3 |

# RESULT

## Item-based

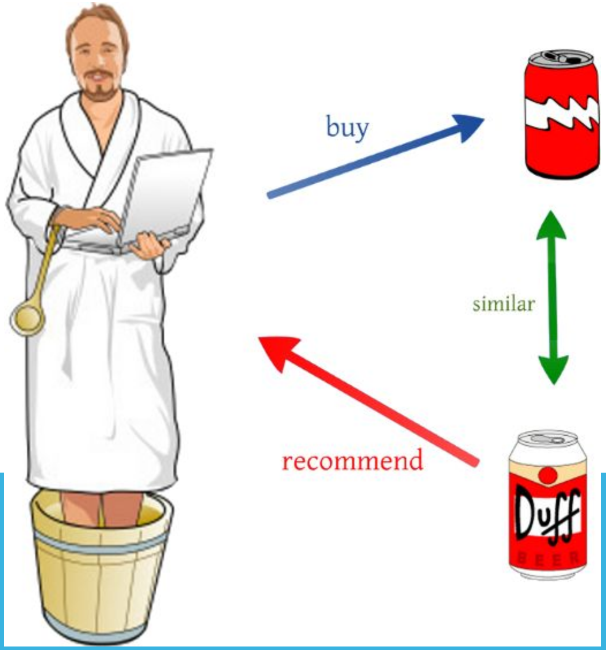
```
['Bonjour Bakery & Deli',
 "Marie Callender's",
 'Sai India Curry',
 "Kitty's Cafe",
 'Jalapeno Grill',
 'Diho Supermarket',
 'China Mama Restaurant',
 'Red Robin',
 'Taqueria Los Parados',
 'Whole Foods Market']
```

## Model-Based

```
name
Pamplemousse Le Restaurant    0.972926
Go Raw Cafe                   0.508439
Spago                         0.505901
Samosa Factory                0.504681
Baladie Café                  0.504628
Ichiza                        0.502571
El Pollo Loco                 0.064086
Sushiko                       0.061469
Binion's Hotel & Casino       0.057244
Rincon De Buenos Aires        0.053358
```

# CONTENT-BASED FILTERING

**Assumption:**
If you like an item in the past, you're going to love a item with the same characteristics.



# APPROACH

**Recommend similar item that user liked in the past based on  properties of items.**

**Methods:**

- Built a item profile and a user profile
  - Type of restaurant
- Recommend items that are similar to those a user like in the past.
  - Dot product
  - Linear regression

# PROFILES

**Restaurant Profile**

- Use categories of a restaurant

```
["['Wine Bars', 'Bars', 'Restaurants', 'Nightlife', 'Italia
n']"]
["['Pizza', 'Restaurants']"]
```

**User Profile**

- Distribute it star ratings over the features and take the average of all ratings per user to get a user profile.

For example,

| | low-abv | high-abv | IPA | Stout | Pilsner | rating |
|---|---|---|---|---|---|---|
| beer 1 | 1 | 0 | 0 | 0 | 1 | 2.0 |
| beer 2 | 0 | 1 | 1 | 0 | 0 | 4.5 |

...will become...

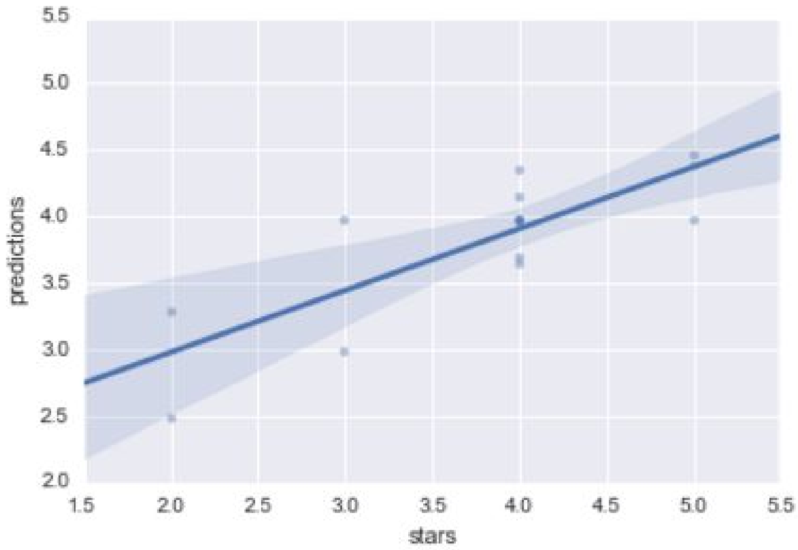
| | low-abv | high-abv | IPA | Stout | Pilsner |
|---|---|---|---|---|---|
| beer 1 | 2.0 | 0 | 0 | 0 | 2.0 |
| beer 2 | 0 | 4.5 | 4.5 | 0 | 0 |

# PREDICTIONS

## Dot Products

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \bullet \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = a_x.b_x + a_y.b_y + a_z.b_z$$

## Linear Regression



# RESULTS

## Dot Product

| name | |
|---|---|
| Big Dog's Bar and Grill | 0.540541 |
| Kabob Palace | 0.540541 |
| Little Bangkok Thai Restaurant | 0.513514 |
| Sakun Thai | 0.513514 |
| Thai Spice | 0.513514 |
| Go Raw Cafe | 0.513514 |
| Jean Philippe Patisserie | 0.486486 |
| Auntie Anne's | 0.486486 |
| Dick's Last Resort | 0.486486 |
| Sesame House | 0.486486 |

## Linear Regression

```
name
Red Velvet Cafe           4.662425
Sandbar                   4.540951
My Buddy's Greek Cafe     4.485396
Palio Pronto              4.475701
Cafe Belle Madeleine      4.475701
Picasso                   4.471745
Elysium Internet Cafe     4.463551
Studio Café               4.456188
Sidewalk Cafe             4.456188
Grand Café                4.456188
Name: predictions, dtype: float64
```

# CONTENT? COLLABORATIVE?

**Use the review text to map the feature of the restaurant.**

- using frequency-inverse document frequency
  - A statistical measure of how import a word is to a document in a collection of corpus

**Recommend business based on similar text review using cosine similarity.**

$$\begin{aligned} \mathrm{idf} &= -\log P(t|d) \\ &= \log \frac{1}{P(t|d)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

# RESULT

```
text[0]
```

'I like Chianti, the outdoor seating area is nice during the spring and fall, while the inside typifies an average Las Vegas dinner spot. The staff is attentive. Entrees are tasty, I liked the beef carpaccio appetizer, as well as the nice, simple pasta dishes and pizzas, and the cioppino is (surprisingly) pretty good too (although the tomato broth is a little overwhelming and the seafood is obviously not coastal fresh). The bread/crostini is horrible (ala white Wonder Bread). Decent wine list, but less than helpful staff in that regard. Overall, I would recommend this spot to anyone stranded in Vegas and craving a bowl of cioppino (p.s., if you know of a better spot for cioppino in Vegas please let me know).'

```
text[134085]
```

'Had the Cioppino and it was great!  Service was very good too.  I would go back again.'

```
text[36686]
```

"We went to a Christmas Eve Dinner with the family and I was let down hard! The Cioppino that I ordered for $50 was far from average. I've had a great cioppino and was not even worthy being called cioppino. The broth was like it was out of a can and the seafood was fishy (not fresh), more like frozen crap. Let's just say the entire dinner was a 1 star rating. The bill was over $500 and for what? I expected so much better. A big let down!!!"

```
text[41471]
```

"Can't say enough about this place.  Had dinner here on 5/11/13, and it was wonderful. Try the Cioppino, it is Excellent."

# EVALUATION

- There is no way to mathematical way to evaluation these models.
- Can follow up with user if they actually go to these recommended restaurants.
- How useful:
    - For vendor, expose their products.
    - For customer, product that they have not experienced.

# Q & A