Developing a Smart Restaurant Recommendation System for Dropbox

AI Studio Final Presentation

Break Through Tech Boston @MIT
December 8, 2023

Introductions

Meet Our Team!

Maame Andoh
Simmons University

Maura Anish
Wheaton College

Miraya Gupta
Wellesley College

Sabrina Lu
Wellesley College

Sandy Zheng
Wellesley College

Vivian Kwong
UMass Boston

Our AI Studio TA and Challenge Advisors

David Fang
AI Studio TA

April Liu
Challenge Advisor

Ameya Bhatawdekar
Challenge Advisor

Presentation Agenda

1. Project Overview
2. Data Understanding & Preparation
3. Modeling & Evaluation
4. Concluding Thoughts
5. Questions

AI Studio Project Overview

“Build an AI model that utilizes machine learning techniques to generate personalized restaurant recommendations for users.”

Our Goal

1. Learn about Collaborative and Content-Based Filtering
2. Create one model using Collaborative Filtering and another using Content-Based Filtering
3. Combine our two models in a hybrid approach to generate high-quality recommendations

Recommendation Systems

- Content-based filtering: predicting what a user would like based on their past preferences
- Collaborative filtering: predicting what a user would like based on the preferences of similar users
- Examples: Netflix's movie and TV recommendations, Spotify's music recommendations

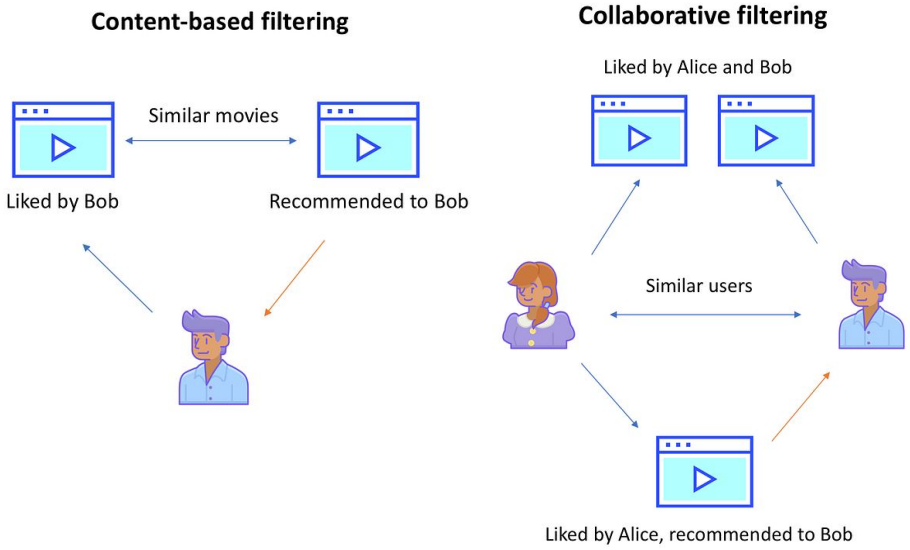
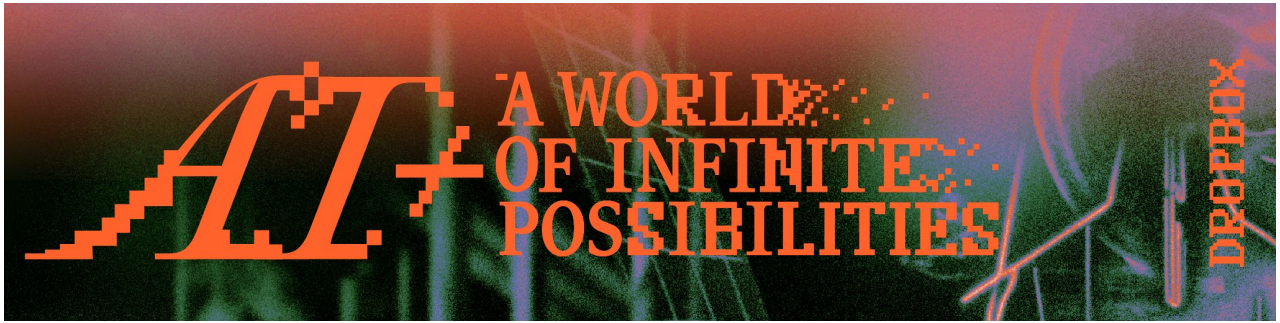


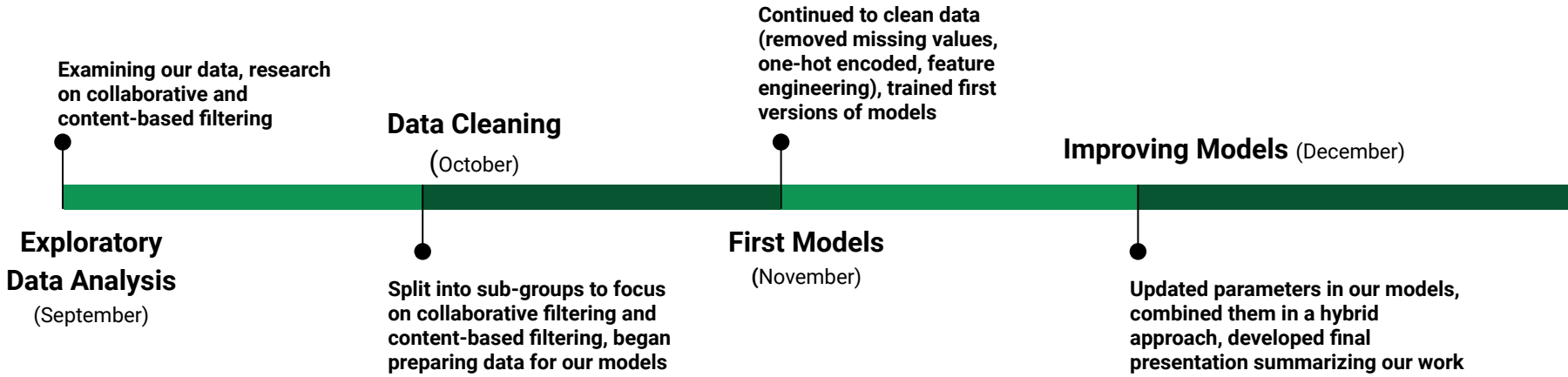
Image credit: Ubajaka CJ (source)

Business Impact

- Dropbox is “laser focused on expediting the creation and implementation of AI-enabled products.”
  - They employ recommendation algorithms to create a list of files/documents that are “Suggested from your activity.”
  - They also use ML to power their “smart move” (file to folder) recommendations.



Our Approach



Resources We Leveraged

To Learn:

- Google for Developers
- YouTube
- Medium
- ML Textbooks

To Code:

- Google Colab
- Python
  - Scikit-learn
  - NumPy
  - Pandas

Data Understanding & Data Preparation

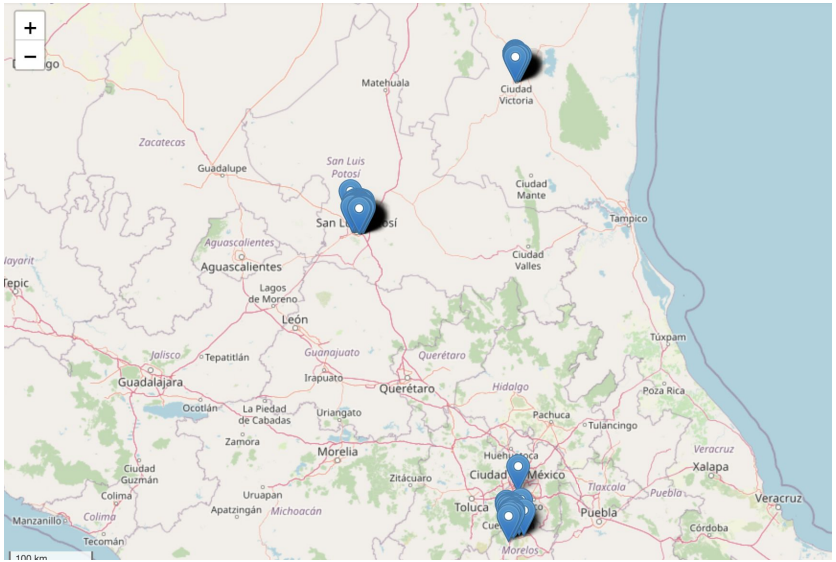
Dataset Overview

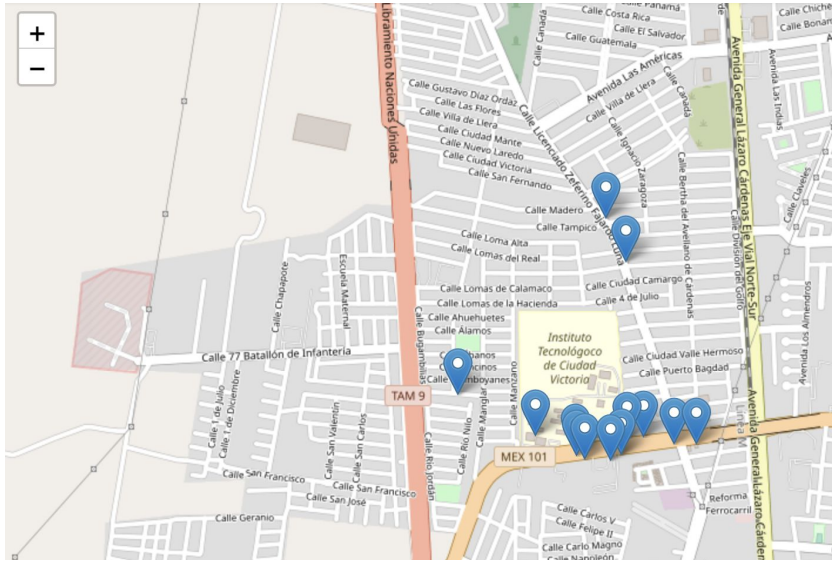
"Restaurant Data with Consumer Ratings" on Kaggle:

1. Restaurant Accepted Mode(s) of Payment
2. Restaurant Type(s) of Cuisine
3. Restaurant Hours
4. Restaurant Parking
5. Restaurant Profiles
6. User Preferred Mode(s) of Payment
7. User Preferred Type(s) of Cuisine
8. User Profiles
9. User Ratings

Exploratory Data Analysis

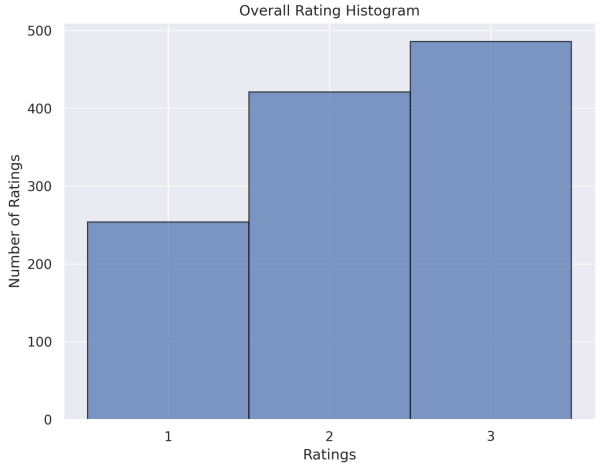
- Examined the amount of rows and unique values each dataset → allowed us to understand what data to use and data processing steps
- Created interactive Maps to visualize the locations of the users and restaurants

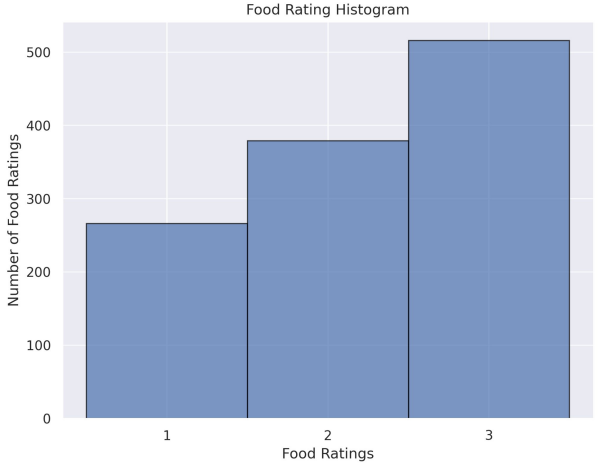


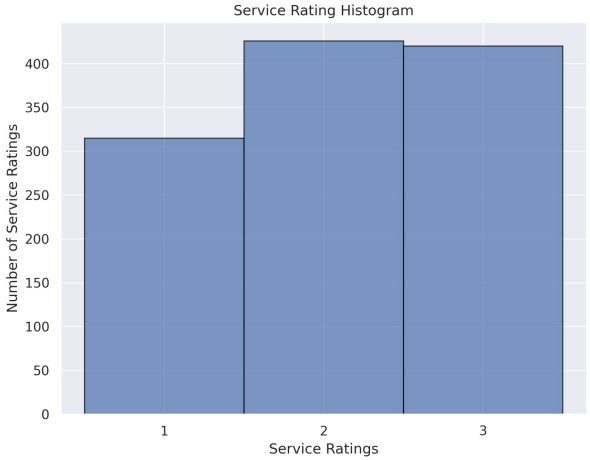


Exploratory Data Analysis

- There were a total of 1,161 user ratings for the overall quality, the food quality, and the service quality of 130 restaurants; we focused primarily on the overall ratings
- Some users rated as few as 3 restaurants, while others rated as many as 18
- Some restaurants had as few as 3 ratings, while others had as many as 36







Data Cleaning - Content-Based Filtering

- We created a new dataset with cuisine type (13 categories), alcohol, ambiance, smoking, and price/budgetary preferences. We chose these features because:
  - Most influential in users' restaurant leanings (in our opinion)
  - Could be found in both user and restaurant datasets and standardized
  - Few to no missing values

restaurant	cuisine types (13)	price	latitude	longitude	alcohol	ambience	dress code	smoking
user	cuisine types (13)	price	latitude	longitude	alcohol	ambience	dress code	smoking

0/1 categorical variables 1/2/3 categorical variables numeric variables

Data Cleaning - Matrix Factorization

- To formulate a matrix using the user ratings, we:
  - Re-scaled the user ratings from 0, 1, or 2 to 1, 2, or 3
  - Filled in all of the ratings for restaurants that users hadn't rated with 0s
  - Created a 138 x 130 matrix (one row per user, one column per restaurant)

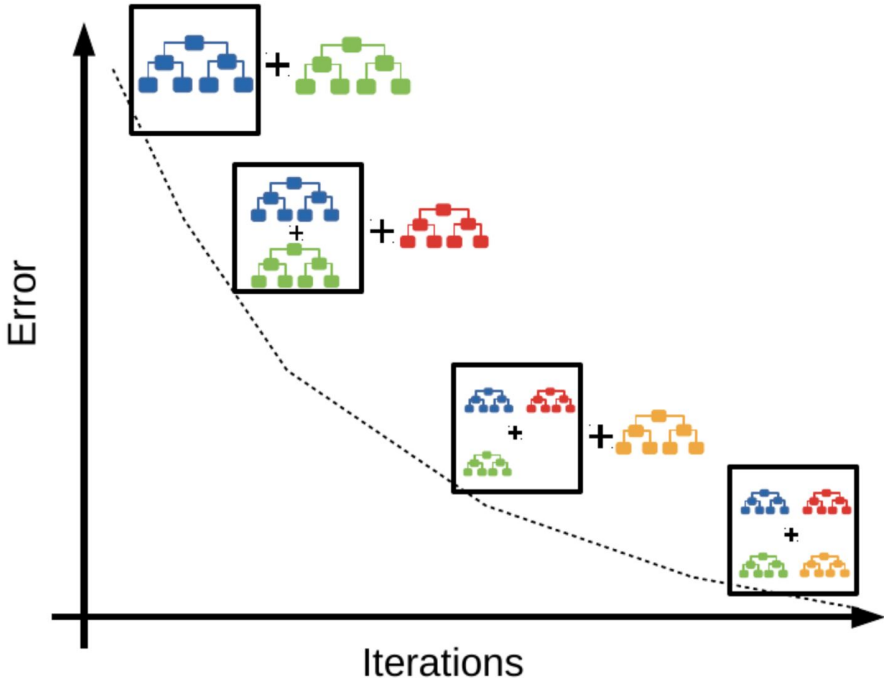
	1	2	3	…	128	129	130
1	[[0.	0.	0.	...	0.	0.	0.]
2	[0.	0.	0.	...	2.	0.	0.]
3	[0.	0.	0.	...	0.	0.	0.]
…	...						
136	[0.	0.	0.	...	0.	0.	0.]
137	[0.	0.	0.	...	0.	0.	0.]
138	[0.	0.	0.	...	0.	0.	0.]]

Only 6.5% non-zero!

Modeling & Evaluation

Model Building - GBDT

- Used a Gradient-Boosted Classification Decision Tree to predict user ratings
- Tuned hyperparameters: learning rate, max depth of tree, number of estimators
- Trained the gradient boosting classifier on the training set, predicted on the validation set and testing set



Model Building - Matrix Factorization

- Factorize our user-restaurant matrix M into 3 matrices: U, D, and V
  - U is the user latent matrix (138 x r)
  - D is the diagonal singular value matrix (r x r)
  - V^T is the restaurant latent matrix (r x 130)
  - r is the number of latent factors needed to account for the variability in ratings

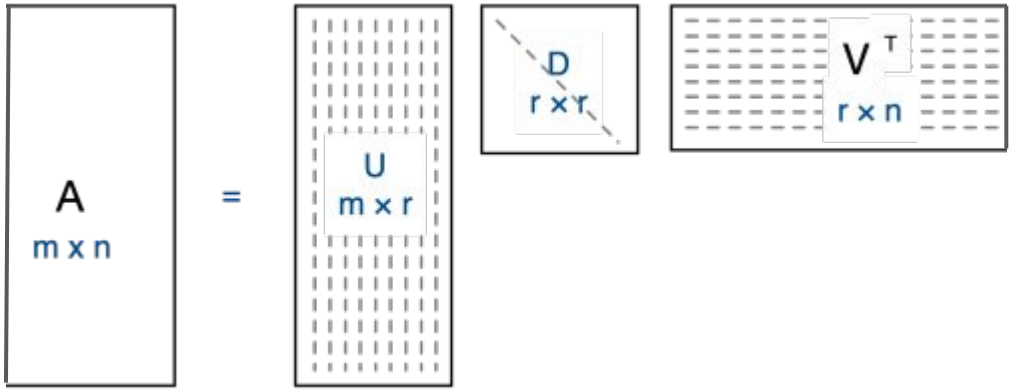


Image Credit: Adel Ahmadyan (source)

Model Evaluation - GBDT

- Training score: 0.839
- Validation score: 0.55
- Testing score: 0.611

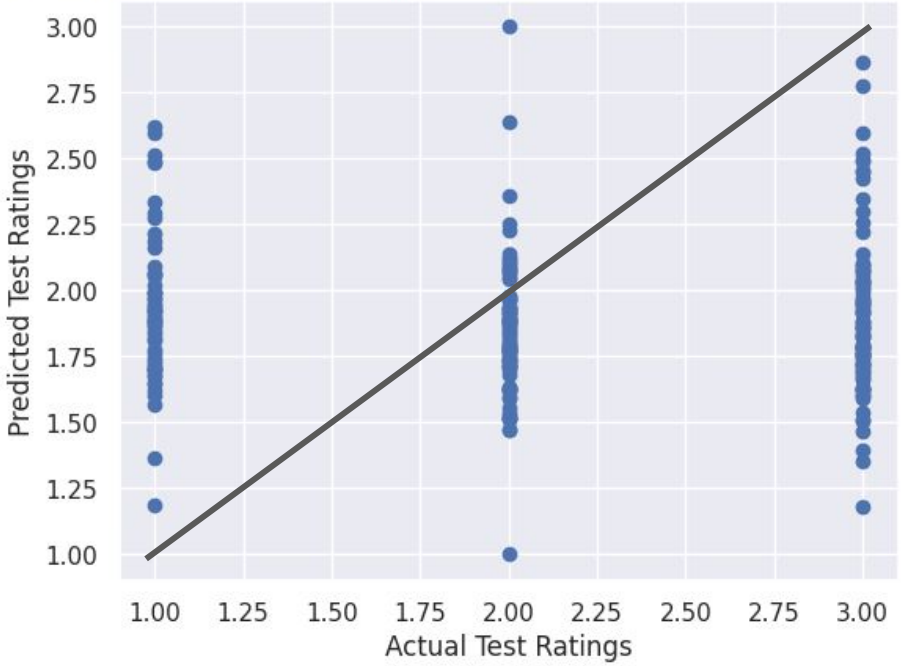
Classification Report:

	precision	recall	f1-score	support
1.0	0.80	0.69	0.74	29
2.0	0.50	0.44	0.47	55
3.0	0.49	0.59	0.54	56
accuracy			0.55	140
macro avg	0.60	0.57	0.58	140
weighted avg	0.56	0.55	0.55	140

Model Evaluation - Matrix Factorization

- We evaluated our first matrix model by removing 20% (232) of the known (1161) ratings, then running SVD and determining how well our model predicted those ratings

Correlation: -0.015

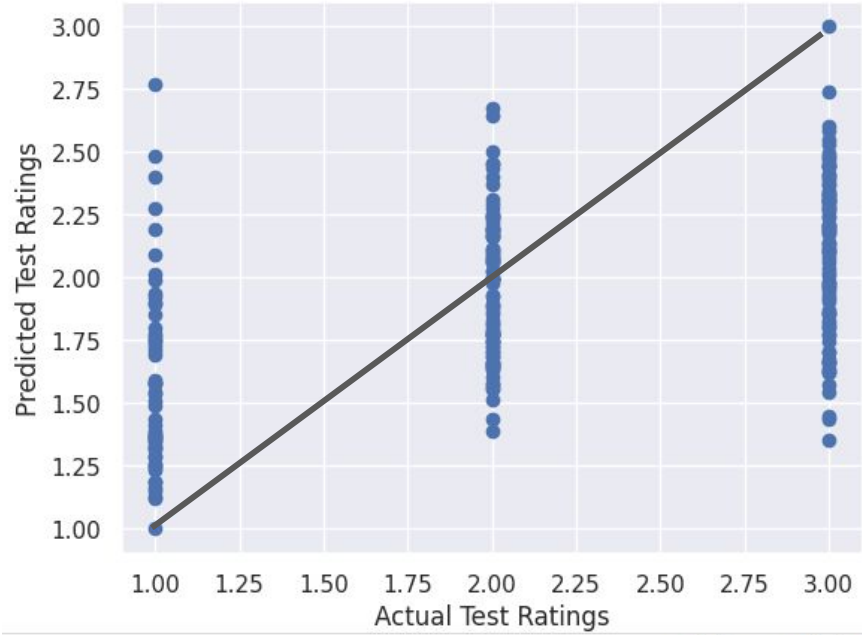


RMSE = 0.87

Model Evaluation - Matrix Factorization

- Then, using a gradient descent with 3000 iterations, a learning rate of .02, and regularization (from "Recommender System – Matrix Factorization" by Denise Chen), we re-computed a prediction matrix for the 929 ratings we left in the training set (increased computation time)

Correlation: 0.50



RMSE = 0.71

Model Combination

- With our improved decision tree and matrix factorization models, we hoped to combine them in a hybrid approach to make one model by averaging the predicted ratings for each user for each restaurant

Restaurant ID	132560	132561	132564	…	135109
GBDT Output	1.4	2.9	1.8	…	2.2
Matrix Output	2.5	2.7	1.2	…	1.6
Avg. Output	**1.95**	**2.8**	**1.5**	**…**	**1.9**

Recommend restaurant 132561 to this user

Insights and Key Findings

Content Based:

- Use of GBDT allowed us to create an algorithm that used the error of previous trees to improve on the performance of following models.
- This produced high training, validation, and test scores. The decreased validation and test scores, could be attributed to limited data.

Collaborative:

- Gradient descent algorithm provided better quantitative results and better restaurant recommendations to users (recommended restaurants with high average ratings)
- Gradient descent had less variation in ratings and recommendations (recommended 40 unique restaurants as opposed to 83 unique)

Final Thoughts

What We Learned

- Difficulty of working with 1) such few ratings and 2) a lack of numeric data
- Rewards of merging datasets with different information spread across them
- Limitations of recommender systems (cold-start problem for new users)
- Difficulty working with a remote team with very different schedules

Potential Next Steps

- Cluster users using their categorical data, then run matrix factorization on user clusters for another hybrid content-based & collaborative approach
- Cluster restaurants using their categorical data, then recommend similar restaurants to those users already rated highly
- Use a neural network discover more complex relationships between users and restaurants


Questions?