Personalized Yelp Recommendations

GA Tech CSE6424 Data Visualization - Team Project

Brandon Tran
John Hermina
Timothy Marx

Problem Statement

- Yelp is a great resource for finding places to eat!
- However, unless you're an avid Yelp user -- Yelp cannot recommend a <u>personalized</u> search.
  - Multiple written multiple reviews,
  - Multiple business visits/ratings
- Heck, Yelp's Recommendations go on for pages and pages like google searches!
  - This isn't personalization more than just recommendations!

Project Goal

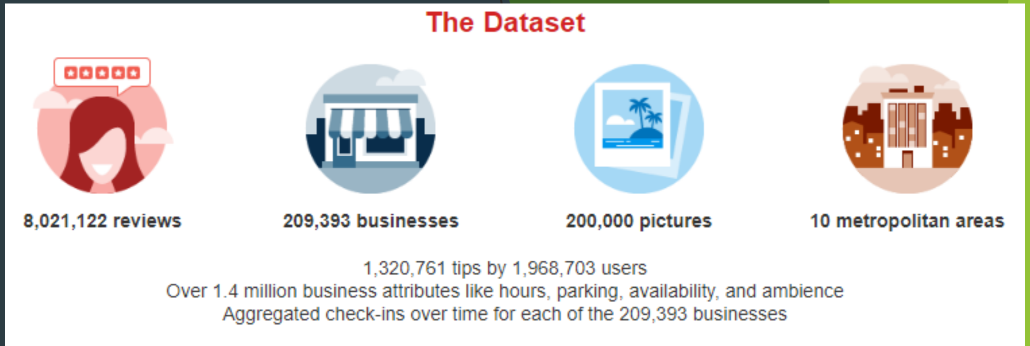
- Create an app that can allow users to input their food preferences & output a personalized restaurant search

- Additionally, provide metrics & limit the results as to allow users to make informed decisions quicker and with confidence!

Recommendation System Details

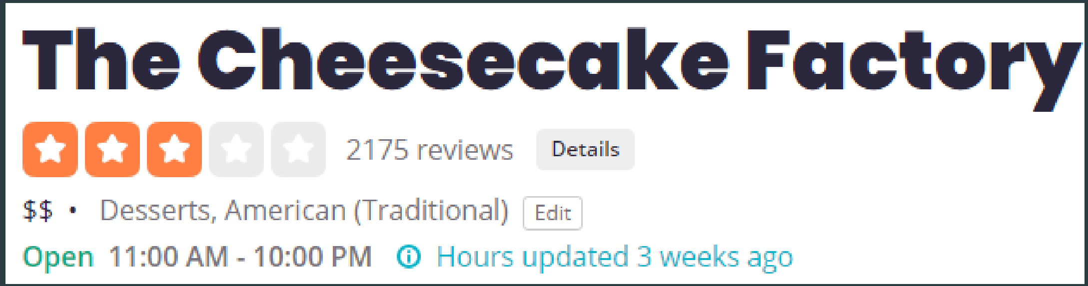
- Package: LightFM
- LightFM maps user/item features and performs dimensionality reduction to find latent factors (see Matrix Factorization) that exist in your model
- The model of choice here was a <u>user-based collaborative filtering</u> model
  - A method that exploits user's features (cuisine preference) to create a model that can be used to predict for a **new user (you!)** based on their own preferences

What exactly are the inputs? How was the model trained?



- The Yelp Challenge Dataset contains lots of information
- Specifically, we used what the business tagged itself as

- Ex: The Cheesecake Factory lists under {Desserts,American (Traditional)}



- These business tags are used as inputs and the LightFM model is trained on what users that exist in the system that have reviewed +3 locations.
- Recommended restaurants are populated based on highest similarity score

Example of how User-based Collaborative Filtering works (Next slide)

Model Input & Output Overview

User_profile = ["Japanese","Thai","Burgers","Korean","Soups"]

Feature Vector for this user:

"Japanese"	"Burgers"	"American (New)"	"Wigs"	"Thai"	……	"Hair Salon"
1	1	0	0	1	…..	0

Sparse Matrix{
Dim:1xN
Values: {("Japanese",1)
…
("Soups",1)
}
N = # unique features

Model.predict(users_id=0,
items_id=
user_features=
)

All_items = {0,1,2,…,M}

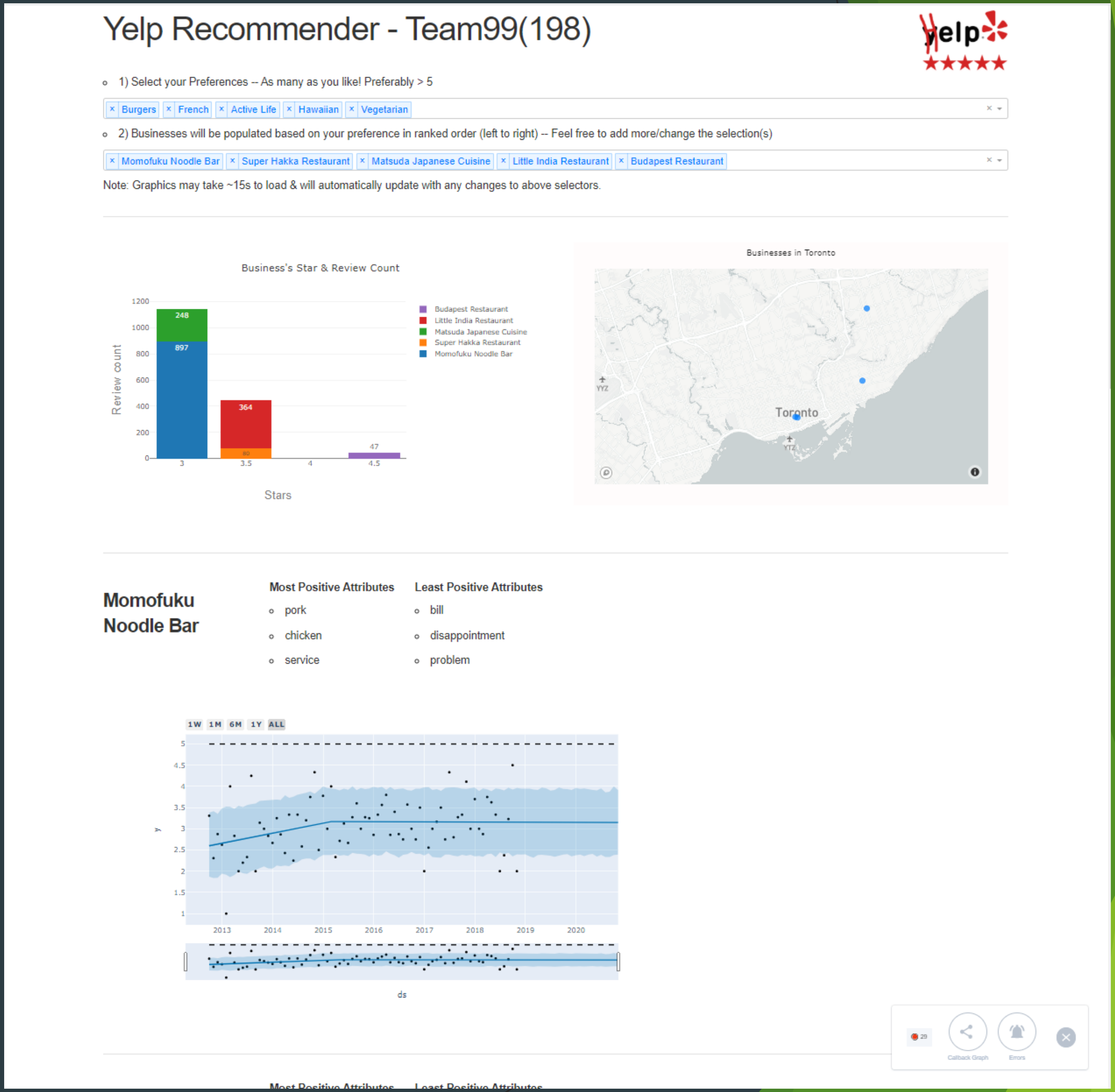
M = # businesses

Sort

Top X recs

Putting it all together!

- 1) Preferences are fed into the model in the 1st drop-down menu
- 2) Recommendations are automatically populated from the LightFM model
- 3) Graphics are also automatically updated once businesses are populated



Packages/Software Utilized

- Python
  - Pandas – Dataframe structure
  - Numpy – General computation
  - LightFM – To compute the Collaborative Filtering Algorithm
  - Dash/Plotly – To generate the dashboard & Callback functions
  - Mapbox – To generate the map
  - FBProphet – To generate the forecasts
- Docker
  - To instance and package the project

How this can be improved

- Input options could be limited/streamlined
  - “American (New)” is likely the same as “American”
    - Also this blanket term may encapsulate other input terms such as “Burger” or “Pizza” which adds a level of confounding --- Maybe remove generic cultural words and allow the individual food items only
  - “Hair Salon” and “Wigs” are not food!
    - Manual check of keywords/tags, although conducted, was not sufficient at removing all nonsense entries
- The dataset could be expanded
  - For the project, we focused only on the Toronto area to keep the project form-fit for the short timeline we had