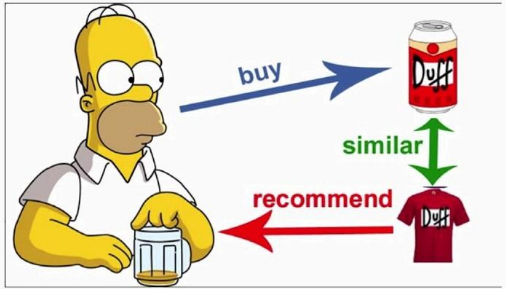
Yelpify

Group Members: June Yang, Quoc Dung Cao, Runqiu Hu, Zichen Liu

2020-12-16

Background



- The Yelpify team builds a recommender system that make recommendations for users and merchants.
- How it works:
  - Given a user, the system recommends a desired number of businesses that the model thinks the user will like;
  - Given a business, the system recommends a desired number of users that the model thinks will most likely favor this business.
- Special about Yelpify:
  - We make recommendations for both existing and new users (and vice versa for businesses);
  - Model is generalizable to a wide range of applications, such as movie recommendation, or hotel recommendation.
  - Strong model performance: powerful and fast

Data: Yelp academic datasets

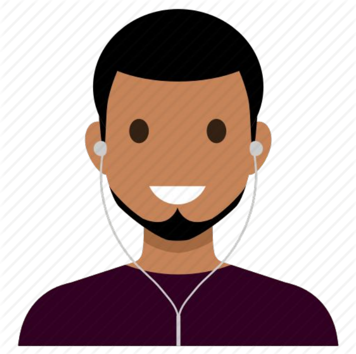
- Review.json
  - business ID, review ID, user ID, text of review, rating
- Business.json
  - features about the business, e.g. location, name, whether open or not
- User.json
  - features about the user, e.g. count of reviews, average ratings given
- Data processing:
  - Join by user ID and business ID
  - Clean null and duplicate values
  - Reduce size

Use cases

For a user:

“I’d like to know what to eat based on my past reviews.”

“I’d like to know what to eat based on my demographics.”



For a business:

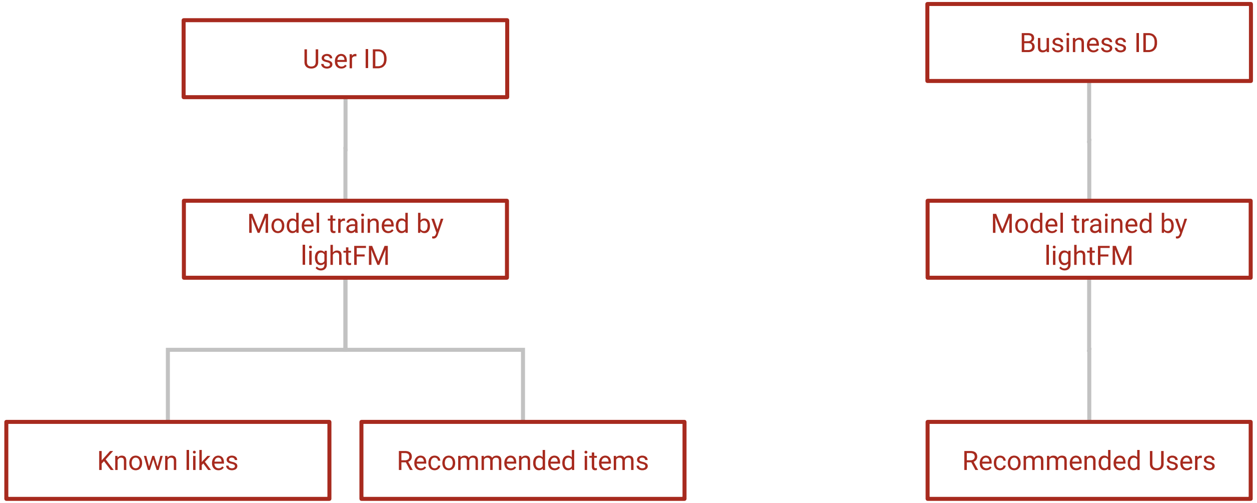
“I’d like to know who like my food most.”

“(optional) I’d like to know what features of my business will attract clients most.”

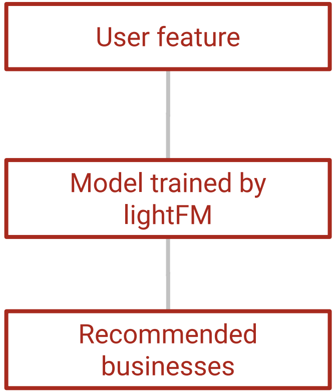
Project Directory Specification

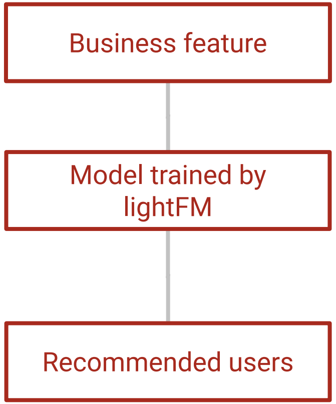
```
yelpify/
├── LICENSE
├── README.md
├── codebase
│   ├── __init__.py
│   ├── codebase.py
│   ├── data
│   │   └── test.csv
│   ├── tests
│   │   ├── __init__.py
│   │   ├── test_codebase.py
│   │   ├── test_data_pre.py
│   │   ├── test_model_cf.py
│   │   ├── test_model_hybrid.py
│   │   ├── test_recommend_hybrid.py
│   │   └── test_recommend_known.py
│   └── version.py
├── doc
│   ├── Design_Functional.md
├── example
│   ├── recommendation_system_cf.py
│   └── recommendation_system_hybrid.py
├── requirements.txt
├── setup.py
└── yelpify
    ├── __init__.py
    ├── data_preparation.py
    ├── model_cf.py
    ├── model_hybrid.py
    ├── recommend_hybrid.py
    └── recommend_known.py
```

Component Specification: Recommendation for existing users/businesses

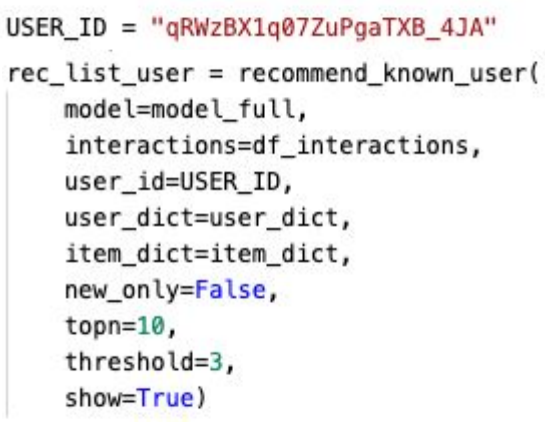


Component Specification: Recommendation for new users/businesses

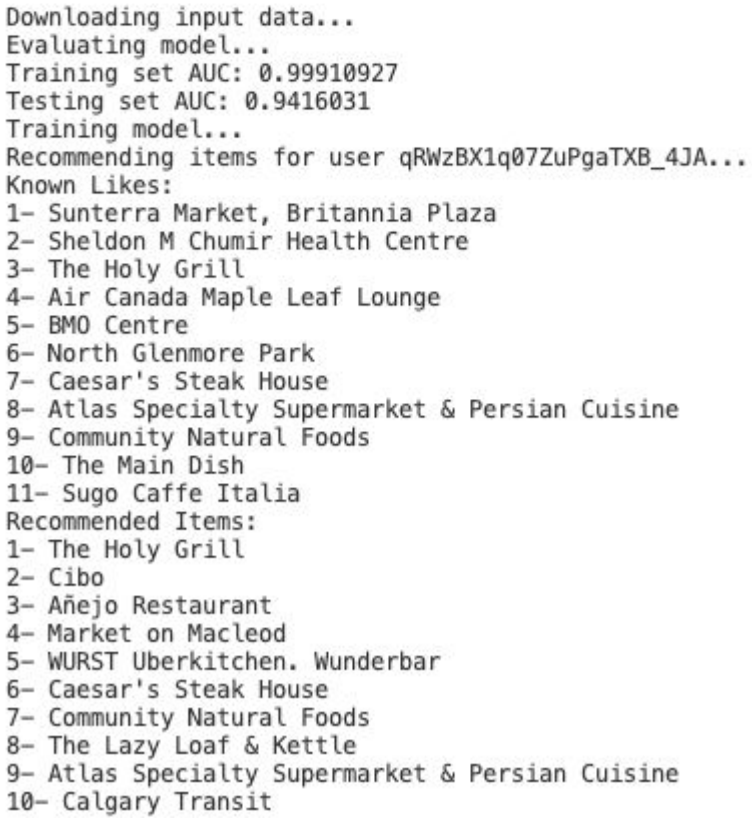




Code Example



```python
USER_ID = "qRWzBX1q07ZuPgaTXB_4JA"
rec_list_user = recommend_known_user(
    model=model_full,
    interactions=df_interactions,
    user_id=USER_ID,
    user_dict=user_dict,
    item_dict=item_dict,
    new_only=False,
    topn=10,
    threshold=3,
    show=True)
```



```
Downloading input data...
Evaluating model...
Training set AUC: 0.99910927
Testing set AUC: 0.9416031
Training model...
Recommending items for user qRWzBX1q07ZuPgaTXB_4JA...
Known Likes:
1- Sunterra Market, Britannia Plaza
2- Sheldon M Chumir Health Centre
3- The Holy Grill
4- Air Canada Maple Leaf Lounge
5- BMO Centre
6- North Glenmore Park
7- Caesar's Steak House
8- Atlas Specialty Supermarket & Persian Cuisine
9- Community Natural Foods
10- The Main Dish
11- Sugo Caffe Italia
Recommended Items:
1- The Holy Grill
2- Cibo
3- Añejo Restaurant
4- Market on Macleod
5- WURST Uberkitchen. Wunderbar
6- Caesar's Steak House
7- Community Natural Foods
8- The Lazy Loaf & Kettle
9- Atlas Specialty Supermarket & Persian Cuisine
10- Calgary Transit
```

Unit Tests

Data preparation

Collaborative filtering

Hybrid filtering

Lesson learned and Improvement

Setup Travis CI early

Optimize runtime of hybrid model on Travis using a small subsample

Repo reorganization

More elaborated README

Style improvement

WHERE?
WHEN? WHY?
WHEN?
WHY?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?
WHEN?
WHO?
WHAT?
WHERE?
HOW? HOW?
WHEN?
WHEN?
WHERE?
WHOSE?
HOW?
WHEN? WHERE?
HOW?
WHICH?
WHOSE?
WHAT? HOW? WHY?
WHO? WHERE? WHAT? HOW?
WHY? HOW?
WHAT? HOW? WHY?
WHO? WHERE? WHAT? HOW?
WHAT?
WHERE? WHICH? WHOSE? WHEN? WHY?
WHY? HOW? WHERE?
WHO? WHOSE?
WHERE? WHAT? HOW?
WHAT?
WHY?
WHO?
HOW?
WHERE? WHAT? HOW?
WHERE?
WHERE? WHICH? WHOSE? WHEN? WHY?
HOW? WHERE? WHY?
WHERE?
HOW?
HOW?
HOW? WHO?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?
WHERE?
WHAT?
HOW?
HOW?
HOW?
WHAT?
HOW?
WHAT? WHERE? HOW?

Team members' scope of work

June Yang: package selection (surprise, lightFM), recommendation using collaborative filtering, examples, demo, software design, technical review presentation

Quoc Dung Cao: package selection (manually built recommender system), final presentation, setup, license

Runqiu Hu: package selection (scikit learn), unit test, readme, software design, project preview presentation

Zichen Liu: built recommendation system using hybrid filtering, examples, demo, readme