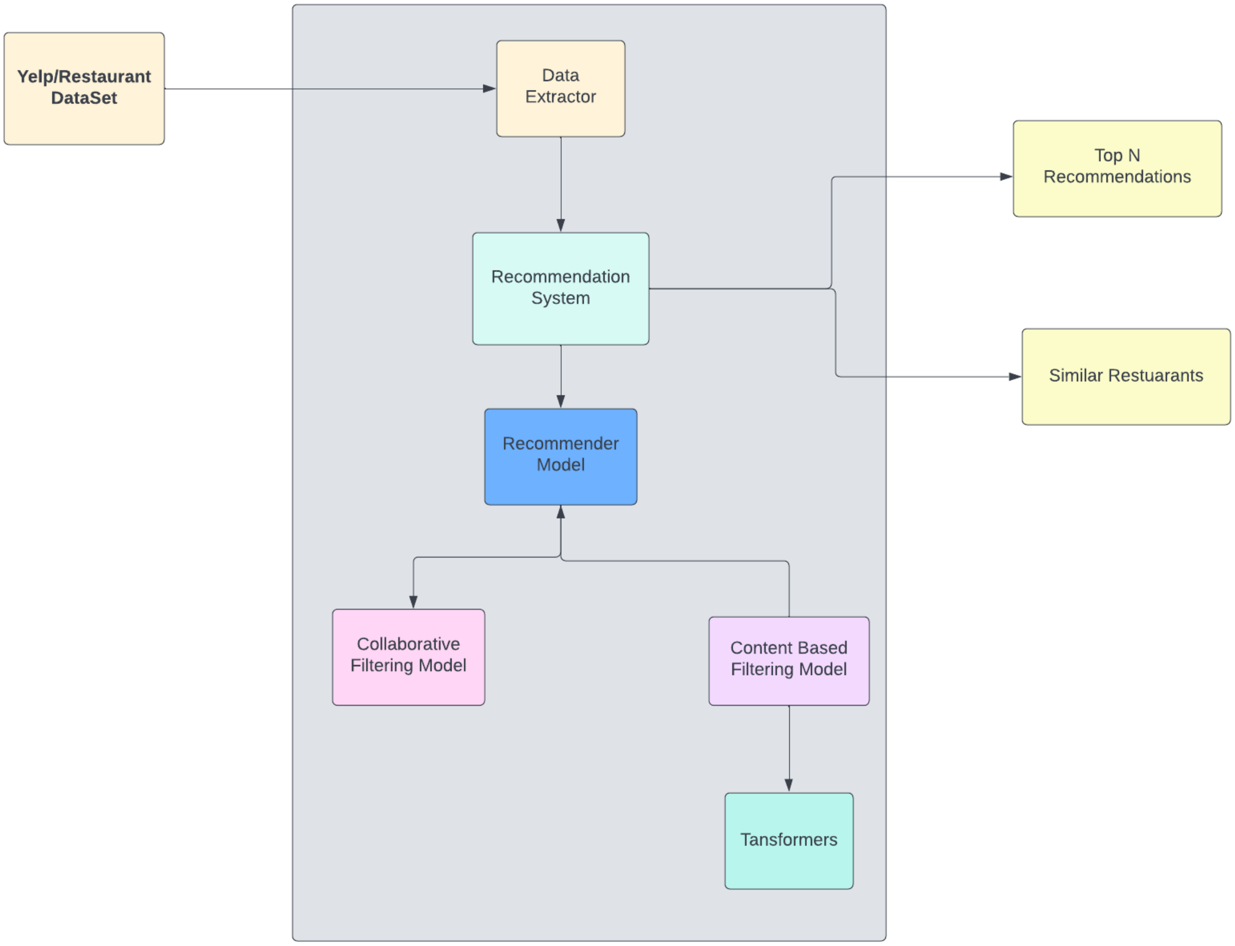
Restaurant Recommender - Component Specification

Group Members: Poornima Muthukumar, Juhi Choubey

Software Components



Data Extractor / Data Cleaning Pipeline: Before utilizing the Restaurant Recommender package, the user has an option to prepare the yelp dataset in required format via the data extractor python module. Due to variability in datasets, this method has appropriate documentation to guide the user on how the raw yelp dataset needs to be fed in and what are the expectations from the raw dataset. The user needs to provide the tar file. The data extractor will process the tar file, extracting 3 json subfiles: user, business and reviews and produce a cleaned csv dataset (*user_business_reviews.csv*) which is written to the path specified by the user.

Recommendation System: This module acts as the interface between the user and the internals of the restaurant recommender package. The users of the package instantiate this module and pass in the required recommendation system that they are interested in. Based on the requested recommender, this module instantiates the recommender model to either a content based model or a collaborative based model. In future if we were to add other recommendation algorithms, we can easily extend this class to support them.

Recommender: This module is the parent module to collaborative and content based recommendation engines. This module contains functions which are common to both the implementations and makes it easier to inherit common functionality such as loading csv and creating data frame, serializing model to pickle file etc.

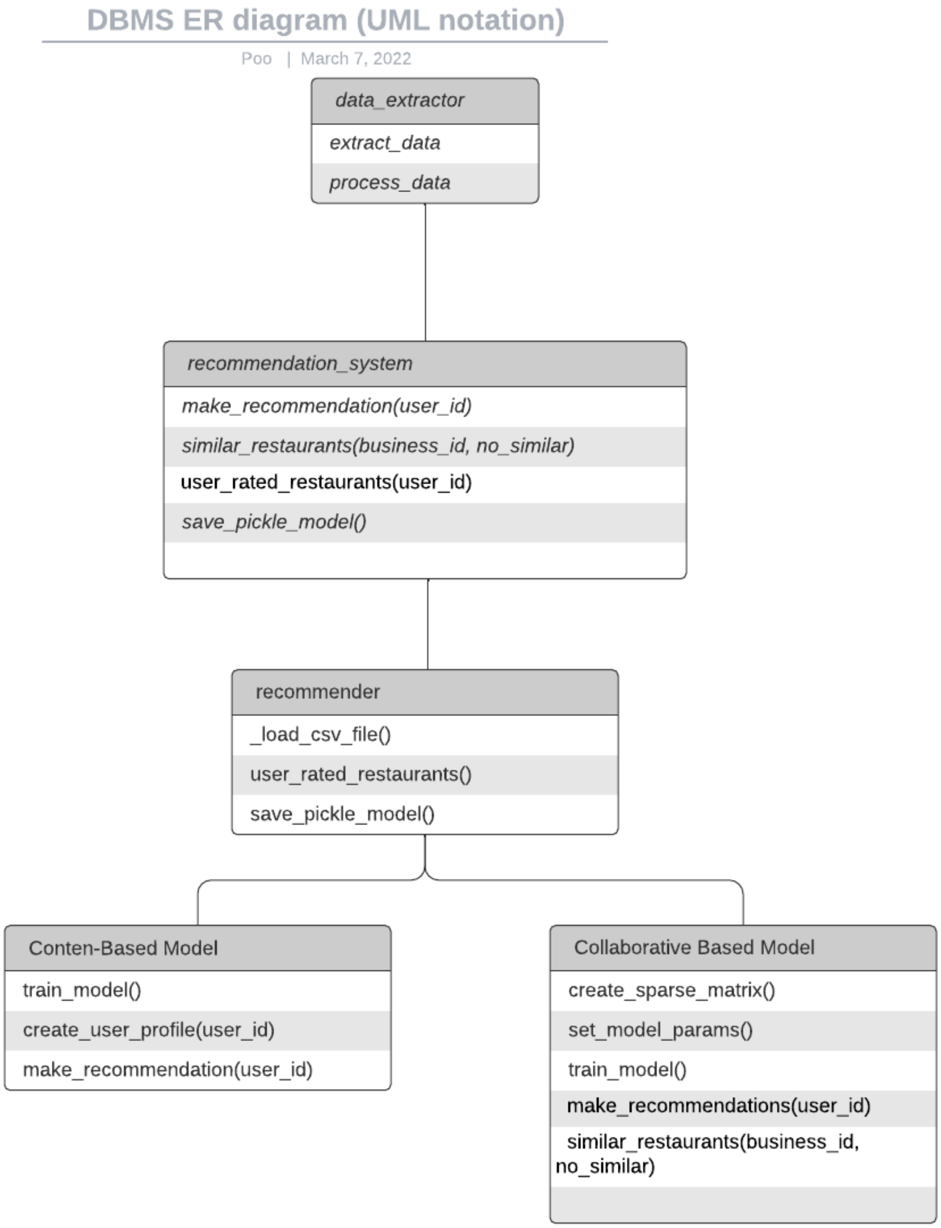
Collaborative Based Recommender Model: This is an internal module that is specific to Collaborative Filtering Model. This model instantiates the collaborative filtering recommendation model based on the alternating least squares method, splits the dataset into train and test and fits the model using train data. It also implements the specific use-cases such as make_recommendation and find_similarity using the model trained to output restaurants that match the filtering criteria.

Content Based Recommender Model: The other internal module that contains the content based filtering recommendation engine. This model instantiates the content based filtering recommendation model utilizing the FeatureUnion estimator from sklearn library. This estimator applies a list of transformer objects in parallel to the input data, and then concatenates the results. It then implements a specific use-case to *make recommendations to a particular user* using the model trained to provide restaurants that match the filtering criteria.

Interactions -

1. The user imports the package restaurantrecommender.
2. The user instantiates a class of the Recommendation System, passing in the csv file and the required recommender type (content / collaborative)
3. As part of the instantiation the model is fit to the train data set.
4. The user then calls the make_recommendation function to get a list of recommendations for the user.
5. The user then calls the similar_restaurant function by passing in the restaurant id and the top N results to return.
6. Finally the user can save the trained model by serializing it to a pickle file by calling the save_pickle_model function.

UML Class Diagram



DBMS ER diagram (UML notation)

Poo | March 7, 2022

data_extractor
extract_data
process_data
recommendation_system
make_recommendation(user_id)
similar_restaurants(business_id, no_similar)
user_rated_restaurants(user_id)
save_pickle_model()
recommender
_load_csv_file()
user_rated_restaurants()
save_pickle_model()
Conten-Based Model
train_model()
create_user_profile(user_id)
make_recommendation(user_id)
Collaborative Based Model
create_sparse_matrix()
set_model_params()
train_model()
make_recommendations(user_id)
similar_restaurants(business_id, no_similar)

Preliminary Plan:

Task	Complete By	Status
Write function requirements	02/12/2022	Complete
Write component specifications	02/12/2022	Complete
Set up github project	02/12/2022	Complete
Write a recommendation system and set up templates for content based and collaborative based models.	02/18/2022	Complete
Train, test and validate the model.	02/18/2022	Complete
Write unit test	02/20/2022	Complete
Write examples	02/20/2022	Complete
Set up python package using set up files	02/22/2022	Complete
Set up requirements.text file	02/22/2022	Complete
Set up CI using travis	02/25/2022	Complete
Update github readme	03/07/2022	Complete
Make presentation slides	03/07/2022	Complete