Free Topic: Food Recipe Recommender System

Team Members

1. Lee Kar Heng Nikolas Basil (Captain)
    - nblee2@illinois.edu
2. Mochammad Dikra Prasetya
    - mdp9@illinois.edu
3. Tanmoy Mishra
    - tanmoym2@illinois.edu

Description

For this project, we chose the task of creating a Recommender System for food recipes. From a project perspective, it would be interesting to go from **scraping real world data** (from a recipe platform with millions of users and recipes) to **training a model** to **evaluating it against an approach detailed in class**.

In general, we think it is also interesting and appropriate for the current situation where people are spending more time staying home and likely looking to pursue/get better at home cooking. Being in this position ourselves, we realise that finding a variety of recipes can be daunting. While we may have a sudden inspiration to cook a specific dish (e.g. lasagna) and are able to find such a recipe through pull based text retrieval methods like the search engine, finding more recipes after that point might be difficult. To tackle this problem, we aim to rely on the strengths of a push based approach such as recommendations to alleviate the decision fatigue and steer users to similar recipes that they are likely to enjoy (e.g. mac & cheese / pasta).

Overview of approach

1. Scrape recipes on this site that have some number of reviews: https://www.allrecipes.com/recipe/281306/lime-ginger-chicken-kabobs-with-peanut-sauce/
    a. For each recipe, scrape data including the users who gave a rating and the actual rating they gave. From there, we are able to find other reviews that this user also rated. We can then add this to our list of recipes to scrape.
    b. This will help build an interaction dataset where each review is rated by multiple users and each user has rated multiple recipes.

    c. Many users provide a detailed description of their review, explaining their rationale for the rating. If time permits, we can use this as user features when scoring future recipes for a particular user or we can mine this information to learn more about a given recipe and generate new features.

2. Scrape metadata for each recipe to be used as features:
    a. Things like preparation technique
    b. Ingredients used etc
    c. Maybe even images if we want to do some CV based recommendations

3. Prepare dataset from the above two steps
4. Setting up a baseline (e.g. basic collaborative filtering)
    a. We do something similar to what we've learnt in class:
        i. For a user and recipe pair, average the ratings of that recipe from similar users.
        ii. Similar users are defined by a similarity threshold, with previous interactions used to calculate similarity.
5. Feature engineering
    a. User Features: We can learn something about a particular user given the user's past reviews.
    b. Recipe Features: We can learn something about a particular recipe given what users have said about it as well as the contents of the recipe itself.
6. Model Training
    a. We aim to predict the rating a user will give to a recipe
7. Model Evaluation
    a. Some data will be held back for evaluation.
    b. We will compare the predicted vs actual ratings.

Tools

1. Selenium (scraping)
2. Python & Pandas (data manipulation)
3. Tensorflow (Recommender System)
4. Numpy / Scikit Learn (evaluation)

Datasets

We will create our own dataset by scraping www.allrecipes.com. The raw dataset, final dataset as well as code used to scrape and process the data will be made available. This dataset is expected to be rich in features and large in quantity
https://expandedramblings.com/index.php/allrecipes-facts-statistics/

Expected Outcome

We will have a rich dataset to train a good recommender system as well as evaluate on. We aim to beat the baseline approach taught in class.

Evaluation

RMSE: Actual ratings vs Predicted ratings
NDCG: Rank recipes by ratings and calculate NDCG based on actual ratings.

Both evaluation approaches should be able to be applied on the baseline and actual approach to enable comparison.

Programming language

Python

Expected Workload

We anticipate that we will easily spend more than 20*3 = 60 hours on the project due to the richness of the dataset allowing us to implement many techniques on it as well as the lengthy process likely required to obtain such a dataset.

A rough breakdown of the work is as follows:

1. Scraping (Total 15 hours)
    a. Scraping reviews & ratings (7.5 hours)
    b. Scraping recipes (7.5 hours)

    We estimate this based on the given time allocated in MP2.1 (4hrs) while providing some buffer given the more complicated nature of the website which likely has measures to deter scraping.

2. Construction of dataset (Total 20 hours)
    a. Basic dataset of user_id, recipe_id, rating (3 hours)
    b. Construction of features
        i. Exploratory Data Analysis (7 hours)
        ii. Feature engineering (10 hours)

3. Setting up baseline approach (5 hours)

4. Model training (10 hours)
    a. Creating general model architecture

    b. Tuning of hyperparameters
    c. Others (e.g. dropout, l2 normalization, embedding layers etc.)

5. Model Evaluation (5 hours)
    a. Evaluation of baseline & actual model

6. Post analysis (5 hours)

7. Compiling results and writing of report (5 hours)

Any extra time from unexpected surpluses can be channelled into subtasks like sentiment analysis/topic modelling for analysis & insights or used directly as model features.