














nicolemichaud03 /  
**Recipe-Recommender-System**










[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)















 **0 stars**  **0 forks**  **1 watching**  **1 Branch**  **0 Tags**  **Activity**

 Public repository

 main  **1 Branch**  **0 Tags**  

 Go to file  **t** [Go to file](#) [+](#) [Add file](#) [Code](#) [...](#)

 nicolemichaud03	organizing pdf files	now	 
 PDFs	organizing pdf files	now	
 images	finishing up project	4 minutes ago	
 .gitignore	Trying different things for modeling	3 weeks ago	
 Miscellaneous_notebook.ipynb	Created using Colaboratory	2 minutes ago	
 NNnotebook.ipynb	Created using Colaboratory	5 minutes ago	
 README.md	finishing up project	4 minutes ago	

 **README**  

## Recipe Recommendation System with Dietary Restriction Embeddings

It can be hard to continuously come up with new and interesting recipes to cook, especially if you have certain dietary restrictions. Many people use websites such as food.com to find, try, and rate recipes. From user and recipe data from Food.com, can we provide users with recommendations for the next recipes that users should try, taking into account their dietary specifications? Taking into account user preferences, shown by recipes they have rated highly in the past, as well as their certain dietary restriction categories of recipes, we create a model that is able to provide the top 3 recommendations for next recipes to try that the user is most likely to rate highly.



## Business Understanding

The proposed recipe recommendation system would be useful, because it can be time consuming and repetitive for people to come up with new recipes to make all the time, especially when their options are narrowed down due to dietary restrictions. If a recommender system took into account not only a user's preferences, based on past recipes that the user has rated highly, but also the type of diet category that the user is most likely to enjoy a recipe from, then this can simplify this task for the user and in turn, make them favor our service more for finding recipes in the future.

While recommendations based solely off of past user ratings can be good recommendations, adding in more information (in this case, diet-types of recipes) can make the recommendations more personalized. With a neural network, extra user or item features can be incorporated into a recommendation system model easily, in the form of embeddings.

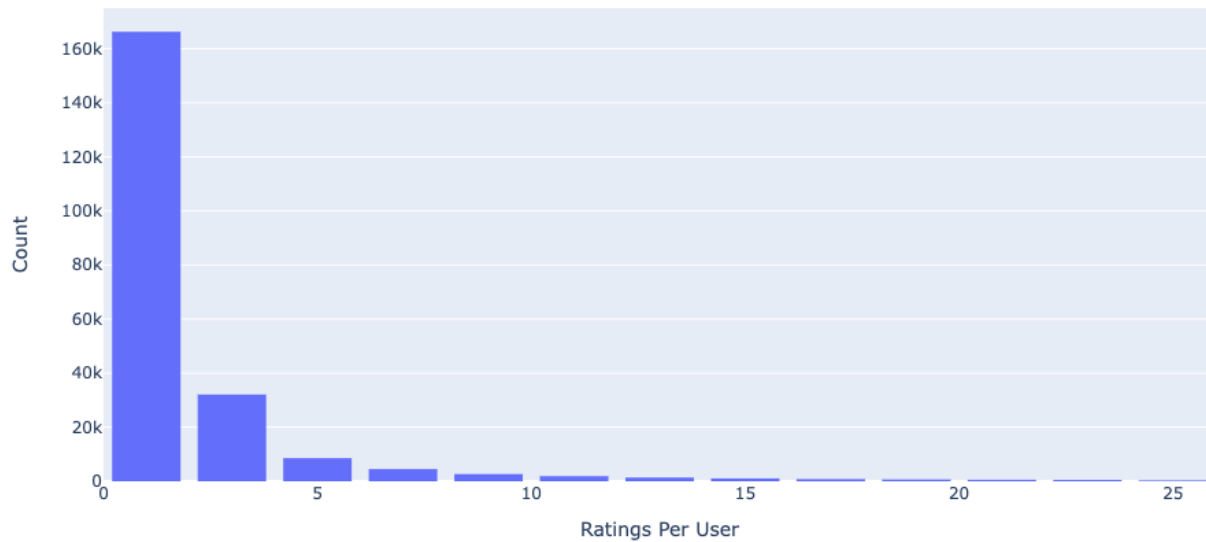
See this [blog post](#) to learn more about the different types of recommendation systems you can build with neural networks.

## Data Understanding

The [user and recipe data](#) is from Food.com, collected from the years 2000 to 2018.

The data contains 226,570 unique users who have rated between 1 and 7,671 recipes:

Distribution Of Number of Ratings Per User (Clipped at 25)



And it contains 231,637 different recipes that have been rated by between 1 and 1,613 users:

Distribution Of Number of Ratings Per Recipe (Clipped at 25)



## Data Preparation

In the miscellaneous notebook associated with this project, the different features of the recipe data were compared to see which ones would contain the most information about each of the three diet types that will be taken into account (vegetarian, vegan, and gluten-free). It was determined that the 'tags' feature contained the most information about diets that were gluten-free, while a different approach, ingredient filtering, would be necessary for recipes of the vegan and vegetarian diet-types.

A text cleaning function removed punctuation, numbers, and symbols from the relevant text data columns, and made all letters lowercase. Using food category information mainly sourced from the [USDA](#), I compiled an extensive list of the ingredient words that would be used in recipes containing meat, seafood, and other animal products. Recipe's that did not contain any of the animal product ingredients were labeled vegan, and recipes that did not contain specifically meat or seafood were labeled vegetarian. Recipes can be all or none of these diet types. A new column was created that combined the designations for each of these diet types for each recipe to be used as a single embedding in the neural network model.

## Modeling and Evaluation

This is essentially a regression task, achieved by Collaborative Filtering. We are trying to predict a user's rating of recipes they have not yet tried, and we are doing so by comparing the user to other similar users and the recipes that they've rated highly (user-user similarity). Then, we want to rank these retrieved item ratings to provide the user with the top "n" recommendations they are most likely to rate highly. For this purpose, we need to use a multi-task neural network model that achieves both the retrieval and ranking tasks.

Using [TensorFlow Recommenders](#), I created an initial (baseline) model that contained only the embeddings for the user IDs and recipe IDs and made recommendations only based on previous rating data from users, and that completed both retrieval and ranking tasks for recommendations. By compiling models with different weights assigned to both the retrieval and ranking task, it was determined that the joint model (both retrieval and ranking tasks given weight) performed the best overall.

The joint model performed better than the ranking-only and the retrieval-only models with a Top-100 Retrieval Accuracy of about 0.001 and an RMSE of about 2.196.

In the next model, I added in an embedding for the diet-type feature that I created.

This model had a top-100 retrieval accuracy of about 0.0023 and an RMSE of about 1.676.

I then used ScaNN (Scalable Nearest Neighbors) in an attempt to optimize the speed and performance of the retrieval task. Compared to brute force serving, ScaNN was more efficient.



### Releases

No releases published

[Create a new release](#)

### Packages

No packages published

[Publish your first package](#)

### Languages

● Jupyter Notebook 100.0%