CSE-158 - Assignment 2
ARAG - All Recipes Are Great

Authors: Pramodya Rajapakse, Adarsh Patel

Task 1. Exploratory Analysis of the Dataset

The dataset we chose was the Food.com Recipes and Interactions, a database of submitted recipes as well as a history of interactions between users and recipes. Each recipe, parsed from RAW_recipes.csv, consists of a name, text description, tags, nutritional information, ingredients, steps, minutes to prepare, the date and user of submission, as well as a unique ID. Each interaction, gathered from RAW_interactions.csv, consists of the recipe and user ID associated with it, a whole number rating between 0 and 5, the date of the rating, and a text review. We decided to split up the interactions dataset into train/validation/test subsets with a split of 50%/25%/25%. A few basic metrics are outlined below.

Total number of interactions	1,132, 367
Total number of recipes	231,637
Average rating of recipes	4.411
Average number of reviews per recipe	4.889
Average number of reviews per user	4.998

Table 1: Basic information about the dataset.

1.1: Distributions on Dataset

Additionally, we computed some statistics and distributions that we thought may be useful indicators to influence our feature engineering. Some of the more intriguing findings are shown here.

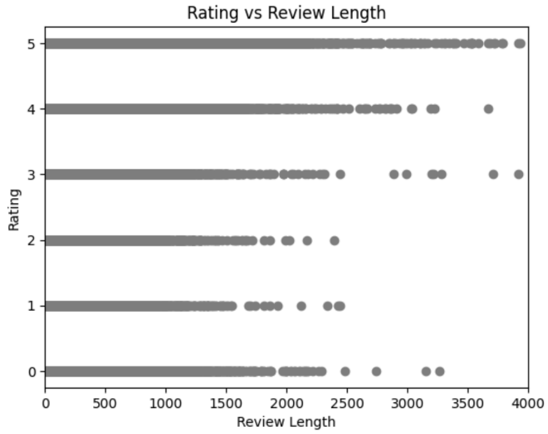

Figure 1: Scatter plot of rating vs review length.

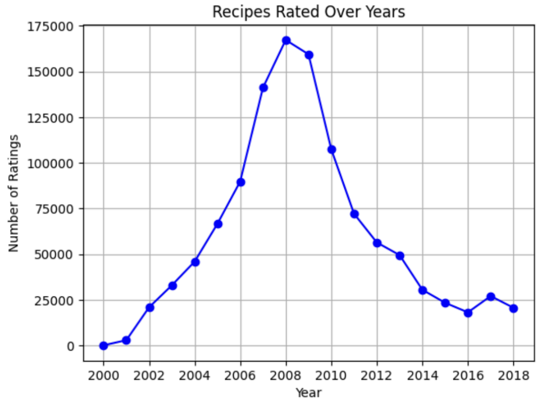

Figure 2: Line graph of recipes rated per year.

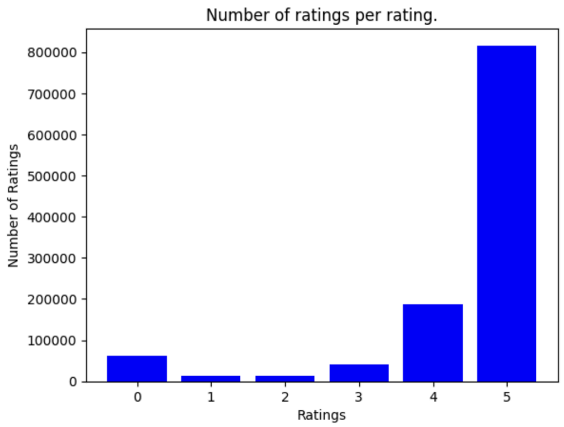

Figure 3: Bar plot of the number of recipes per rating.

1.2: Number of Ratings Per Rating

Of the properties we explored, the count of ratings per class of rating (0-5) was the most significant, as the data (Figure 3) clearly shows a heavy skew towards a high rating of 5 (over 72% of interactions), with the middle ratings of 1, 2, 3 being used very infrequently. This high disparity could be the result of various reasons, like perhaps users are generally more likely to select recipes they already want to make, leading to higher ratings, or users may feel more satisfied as they were the ones who made the dish. Regardless of the underlying reason, we knew this heavy skew could be exploited as a decent baseline predictor when it comes to ratings.

1.3: Number of Reviews Per Year

Also, notice the line graph with the peak number of recipes rated around 2008 then sharply declining until 2018. It is quite interesting that the number of reviews per year would decline so dramatically. This may be due to the fact that food delivery services such as UberEats, PostMates, and DoorDash started coming out around 2010 and caused people to cook less and get delivered more.

The inclusion of text-based reviews also made this dataset a great opportunity for sentiment analysis. We will further discuss this approach in Task 4 in regard to existing literature.

1.4: Common Trends in Dataset

A common trend we noticed was how many of our assumptions behind which attributes and features should be indicative of certain outcomes turned out false. For example, we had assumed that there would be more interactions for shorter recipes, as those would be more enticing. However, we found that there were more than double the number of interactions for recipes longer than 60 minutes compared to less than 60 minutes (161,647 vs 69,990). Hence, it was clear that a model purely based on feature engineering of our design may not capture the most vital underlying relationships within the data, providing motivation for the use of latent factors, which will be discussed in Task 3.

Task 2. The Predictive Task

2.1 Evaluation Metrics and Baselines

We selected the task of predicting the rating a specific user would give to a recipe. The two most relevant metrics for evaluating the performance of our model would be the accuracy, by rounding our predicted rating up to the nearest rating value and marking it “correct” if the predicted class was the same, and the Mean Squared Error, which would be between the predicted rating value and the true rating. We chose to primarily base performance on MSE, even though we aren’t technically predicting a continuous range of values. Our reasoning behind this is that the MSE value can more accurately encode how far off each prediction was (e.g. 3.5 and 3.9 would both round up to 4, giving the same score if the true value was 3 although 3.5 was closer), and also since the MSE should be more useful when it comes to imbalanced datasets, which our one is.

As mentioned previously, given the heavily skewed nature of the interactions dataset, an effective baseline could be found in a trivial model that always predicted a rating of 5. Utilizing this model on our validation set yielded an MSE of 1.97 and an accuracy of 72.025%, the values of which attest to the imbalanced nature of our data.

2.2 Feature Statistics/Unsuccessful Attempts

From there we explored a few regression models with various combinations of features to try and determine which dimensions may be the most useful for prediction. Our initial attempts utilized attributes we obtained through pre-processing the interaction data.

We also decided to include features regarding the properties of the recipes themselves (drawn from RAW_recipes.csv), including the number of steps in the recipe and the number of minutes the recipe takes to prepare.

Model Features	**MSE**	**accuracy**
Feature 1		
Review Length	1.623	72.025%
Feature 2		
Scaled Review Length	1.623	72.025%
Number of Reviews Per Recipe		
Number of Reviews Per User		
Feature 3		
Scaled Review Length	1.600	70.849%
Average Rating Per User		
Average Rating Per Recipe		
Feature 4		
Number of Steps Per Recipe	1.623	72.025%
Number of Minutes Per Recipe		
Number of Ingredients Per Recipe		
Feature 5		
Sentiment Analysis	1.428	64.500%

Table 2: Table of features we use and their respective MSEs and accuracies.

2.3: Describing Each Feature

For our first feature, we decided to start simple and ask does the review length affect the rating? We can see that the accuracy is similar to the baseline, but the MSE is a bit less. We then added to this by scaling the review length by max length and adding in the number of reviews per recipe and the number of reviews each user has given. In other words, adding in more popular users and recipes. However, we achieved the same results. Then, for feature 3, we decided to still include our scaled review length, but add in the average rating per user and the average rating per recipe. In this case, we got a better MSE, but a lower accuracy. In feature 4, we wanted to see if less ingredients, steps and minutes would call for a higher rating. We know, from part 1, this is unlikely, we've achieved the same results as feature 1. For feature 5, we use sentiment analysis on the contents of the reviews focusing on the top 1000 most popular words and again, found the MSE to be better but the accuracy to be lower.

Task 3. Proposed Model - Latent Factors

We proposed the idea of using a latent factor model to better predict ratings. The primary motivation behind this decision was the fact that it was difficult for us to determine which factors may best explain the variability in the dataset, as their inclusion would depend on our innate assumptions about whether or not they would be helpful.

3.1: The Predictor's Equations

We fitted a predictor of the form $rating(user,\ recipe)\ =\ \alpha + \beta_{user} + \beta_{recipe}$ by iterating the set of gradient update equations until convergence. This is a very similar approach to what we were asked to do in Homework 3.

$$\alpha = \frac{\sum_{u,i \in train} (R_{u,i} - (\beta_u + \beta_i))}{N_{train}}$$

Equation 1: Equation to calculate alpha.

$$\beta_u = \frac{\sum_{i \in I_u} R_{u,i} - (\alpha + \beta_i)}{\lambda + |I_u|}$$

Equation 2: Equation to calculate user betas.

$$\beta_i = \frac{\sum_{u \in U_i} R_{u,i} - (\alpha + \beta_u)}{\lambda + |U_i|}$$

Equation 3: Equation to calculate item betas.

Here, $R_{u,i}$ is the rating user u gave to recipe i, I_u is the set of recipes that user u interacted with, and U_i is the set of users that interacted with recipe i. The value of λ is a hyperparameter we later tuned, the results of which are shown later in this section. The value of α we chose to begin iterations with was the global average rating.

3.2: The Results

The hope was that this kind of approach would exploit the latent factors (γ_u, γ_i) that best explain the relationships among the data, whatever they may be, as the model's objective is to minimize the MSE. Our initial use of this approach yielded excellent results compared to previous models, achieving an MSE of 1.04 (with $\lambda = 5$, 5 iterations). From here we continued testing with both larger iteration counts, to hopefully lead to more accurate converged values, and various λ values, checking performance against our validation set. A record of our results are shown below.

λ value	No. of iterations	MSE
5	5	1.04
5	25	1.0399
4	25	0.9897
3	25	0.923
2	25	0.83
1	25	0.6907
0.05	25	0.5109
0.01	25	0.5085
6	25	1.077
10	25	1.182
0	25	0.5083

Table 3: Table showing lamba values with their number of iterations and respective MSEs outputted.

3.3: Tuning and Optimizing the Model

Based on our experimenting, the best model we could implement used a λ value of 0, and the update equations ran for 25 iterations. Although higher iterations after 25 would in theory lead to more accurate converged values, we didn't observe a noticeable change in the MSE that was produced.

Given the optimal λ value, we thought the model would be overfit to the training data, and not generalize well when it came to the test set. However, the model achieved an MSE of 0.495 on the test set, the implications of which we will discuss in the final section.

3.4: Scalability, Strengths, Weaknesses

Scalability is a major strength of this model, as there wouldn't need to be any changes made to how the model runs. Another strength is that this model doesn't rely on feature engineering on our parts, which can be influenced by our innate assumptions about which features may be most important. This benefit also comes with a cost, however, as a key weakness of this latent-factor model is that it doesn't reveal the dimensions that best explain relationships within the data.

Task 4. Associated Literature

4.1: Similar Datasets and Their Use

The use of machine learning pertaining to recipes has been a widely explored topic, although it has been applied in a variety of ways. The Food.com Recipes and Interactions dataset has been a popular collection of data to train and test recommender models given its size and detailed nature. In our research, the other notable dataset revolving around the topic of recipes was a dataset of recipes from AllRecipes.com, although the two were used in different ways. Both were used in explorations of recipe generation, cuisine categorization, and nutritional estimates, however, the Food.com data was used more when it came to personalization and user recommendation.

4.2: Current State-of-the-Art Methods

In terms of recommendation, some novel approaches have been used that highlight some interesting findings. Morales-Garźon et al., 2023 implemented a rating prediction algorithm using a graph neural network by embedding both the review text and nutritional information. Essentially, each node of the graph was characterized by its connections to neighboring nodes, exposing underlying relationships. They discovered that review text was helpful in improving the accuracy of rating predictions. Lin et al., 2014 built a model for recipe recommendation using what they denoted as a content-based matrix factorization model, designed to exploit the latent factors between different recipes, such as their ingredient lists, preparation techniques, and cuisine. They noted how matrix factorization can be more difficult for recipe recommendations than say movie recommendations due to the higher degree of sparseness in the interaction matrix (a recipe will likely have far fewer interactions compared to how many people watch a movie).

4.3: Nutrition and Cuisine Classifying

Of course, given how crucial diets are to our health, there also has been a great deal of research revolving around nutritional estimates and trying to determine which recipes or aspects of dishes are correlated with health benefits. Rokicki et al., 2018 built a regression model using the AllRecipes.com dataset to try and estimate the healthiness of certain recipes, and specifically which features people would think may be good indicators (based on crowdsourcing) compared to what their feature engineering revealed. They found that the most useful indicators were the list of ingredients, followed by directions and preparation type. Al-Asadi and Jasim, 2023 used clustering and an autoencoder neural network approach to recommend recipes by creating connections between users with similar dietary behaviors. Given how our task was to predict ratings, some approaches that exploited similarity between users may have also been a worthwhile addition.

Cuisine classification is also a major topic that we saw the Food.com dataset being used for. Su et al., 2014 used a support vector machine treating the ingredients as features, and using them to classify a recipe into a class of cuisine. They noticed how certain cuisines could be predicted quite accurately, while others couldn't due to overlap between classes. While this topic itself is somewhat different than the prediction of ratings, it highlights the novel ways attributes of recipes could be turned into features

4.3: Conclusions From Existing Work

Overall, much of the conclusions drawn by existing literature matched what we discovered in our experiments. For one, using the text of the reviews was an effective indicator of sentiment, even more so than a regression model built purely on features regarding recipe information. Secondly, many of the most effective models exploited implicit relationships between users and recipes within the interactions, just as we saw the best performance with a latent-factor model.

Task 5. Concluding Discussion

Running our optimized latent-factor model on our test set yielded an MSE of 0.495, performing significantly higher than any other regression model we built on explicit feature engineering or sentiment analysis.

5.1: Using Different Features

Among the various feature representations we experimented with, it was clear that using the review text was an effective tool, mirroring the conclusions drawn by Morales-Garźon et al., 2023. Features such as the number of minutes to prepare, the number of steps, or the number of ingredients were not effective indicators in the way we suspected they would be. Similarly, trying to include the user's average rating or a recipe's average rating did not work too well either, which could be attributed to the heavily skewed distribution of the data.

5.2: Interpretation of Parameters

Something interesting was that when tuning the hyperparameters, a λ of 0 yielded the best MSE on the validation set. In context, this means that the model is essentially not using the regularizer at all, which should lead to a highly overfitted model. Although we should have seen worse performance on the test set, the MSE was surprisingly lower. This could be attributed to the heavily skewed nature of the entire dataset, as even with overfitting, the high frequency of 5 ratings would make the overall distribution of the test data very similar to the training and validation sets.

5.3: Why did Latent-Factor Succeed

We think the latent-factor model worked better than the regression models we implemented since it exploited connections that we couldn't see. Trying to determine which attributes would be decent indicators of a rating is difficult since it relies on our assumptions about what people looking for recipes care about. Not only is this highly individualized, but there are far fewer interactions per recipe than other types of products (e.g. movies, clothes), something Lin et al., 2014 [] touched upon. By letting the model reveal the relationships, we can remove the human bias from the equation. Of course, as mentioned before, an issue with this approach is that we won't be able to determine which features were the maximal indicators, which wouldn't be a problem with other regression models. So while our model works well for the task of predicting ratings, it doesn't provide us with any additional information to make use of in the future.

Works Cited

Morales-Garźon, Andrea, et al. "How Tasty Is This Dish? Studying User-Recipe Interactions with a Rating Prediction Algorithm and Graph Neural Networks." *SpringerLink*, Springer Nature Switzerland, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-3-031-42935-4_9.

Lin, Chia-Jen, et al. "A Content-Based Matrix Factorization Model for Recipe Recommendation." *SpringerLink*, Springer International Publishing, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-3-319-06605-9_46.

Al-Asadi, Ammar Abdulsalam, and Mahdi Nsaif Jasim. "Dietary Behavior Based Food Recommender System Using Deep Learning and Clustering Techniques." *Wasit Journal of Computer and Mathematics Science*, 30 June 2023, wjcm.uowasit.edu.iq/index.php/wjcm/article/view/126.

Rokicki, Markus, et al. "The Impact of Recipe Features, Social Cues and Demographics On Estimating the Healthiness of Online Recipes." *View of the Impact of Recipe Features, Social Cues and Demographics on Estimating the Healthiness of Online Recipes*, ojs.aaai.org/index.php/ICWSM/article/view/15034/14884. Accessed 4 Dec. 2023.

Su, Han, et al. "Automatic Recipe Cuisine Classification by Ingredients: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication." *ACM Conferences*, 1 Sept. 2014, dl.acm.org/doi/epdf/10.1145/2638728.2641335.