CSE158 Assignment 2 - Predicting User Rating For Food Recipes

1 Exploratory Data Analysis

1.1 Introduction

The objective of this analysis is to explore user interactions with recipes and understand the characteristics of recipes available in the dataset. By examining the interactions and recipes datasets, trends in user behavior, recipe attributes, and rating distributions are uncovered. These insights are aimed at building a predictor model that can estimate a user's rating for a food recipe. This model could then be utilized in a recommender system to suggest food recipes to users based on their predicted ratings.

Two datasets are utilized in this analysis: the Interactions Dataset, which contains user ratings and interactions with recipes, and the Recipes Dataset, which includes detailed attributes of each recipe, such as cooking time, ingredients, and nutritional information.

1.2 Datasets Overview

The Interactions Dataset consists of 1,132,367 interactions, spanning from January 25, 2000, to December 20, 2018. It includes data from 226,570 unique users interacting with 231,637 unique recipes. Ratings are heavily skewed towards 5, with the following distribution:

Rating	Number Reviews
0	60,847
1	12,818
2	14,123
3	40,855
4	187,360
5	816,364

This suggests a strong bias toward positive feedback, which makes sense since you are unlikely to choose to cook a recipe which you would knowingly dislike.

The Recipes Dataset contains information on 231,637 recipes contributed by 27,926 unique contributors. Cooking times show a mean of 93.4 minutes and a median of 40 minutes, though an extreme outlier indicates a cooking time exceeding 2 million minutes, likely a data entry error. Nutritional information reveals averages of 473.94 calories, 36.8g total fat, 84.30g sugar, 30.15g sodium, 34.68g protein, 45.59g saturated fat, and 15.56g carbohydrates per serving.

1.3 Data Analysis

1.3.1 Rating Distribution

Ratings are highly skewed toward positive values, with an overwhelming majority of interactions rated 5. Low ratings (0–2) account for a small percentage of the total, suggesting either a tendency for users to rate recipes positively or an under representation of poorly rated recipes.

1.3.2 Cooking Time

Most recipes take under 60 minutes to prepare, with a significant number taking less than 30 minutes. Outliers, such as recipes with cooking times exceeding 2 million minutes, are likely data entry errors. Shorter cooking times align with user preferences for convenience and efficiency, which may positively impact ratings.

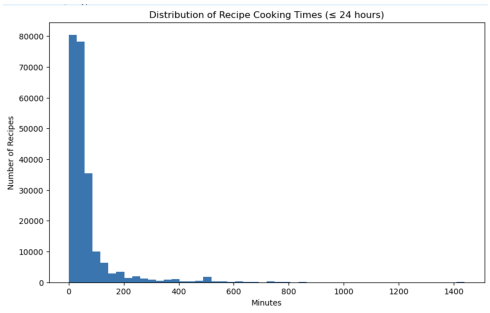

Figure 1: Distribution of Recipe Cooking Times

1.3.3 Number of Ingredients

The majority of recipes use 8–12 ingredients, with very few exceeding 20 ingredients. Recipes with moderate ingredient counts are the most common, reflecting user preferences for recipes that are neither overly simple nor excessively complex. The number of ingredients can influence ratings, as simpler recipes may be perceived as less satisfying, while more complex recipes might demand more time and effort.

1.3.4 Number of Steps

Most recipes involve fewer than 20 steps, with a sharp decline for recipes requiring more steps. Simpler recipes are preferred due to their ease and convenience, while more complex recipes with many steps may negatively impact user engagement.

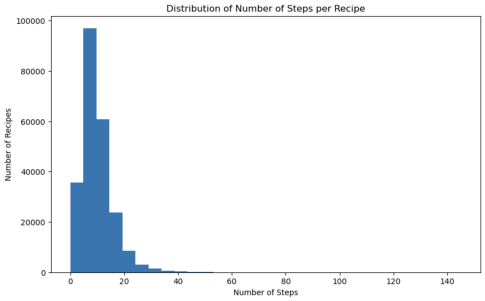

Figure 2: Distribution of Number of Steps per Recipe

1.3.5 Recipe Tags

The most frequently used tags include *preparation*, *time-to-make*, and *course*. These tags are critical for categorizing recipes and improving their searchability for users.

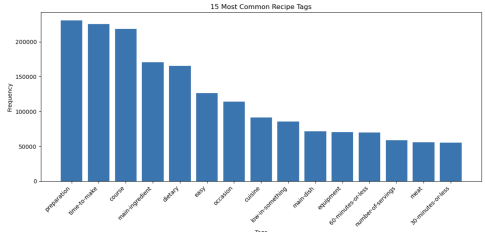

Figure 3: 15 Most Common Recipe Tags

1.3.6 Average Ratings Over Time

Average ratings peaked between 2008 and 2012 but have shown a decline since. This trend could be attributed to changes in platform popularity, shifts in user demographics, or evolving rating behaviors.

1.3.7 Calories

Most recipes contain fewer than 500 calories per serving, with outliers exceeding 1,000 calories. This distribution suggests a focus on low-calorie recipes, likely reflecting the preferences of health-conscious users.

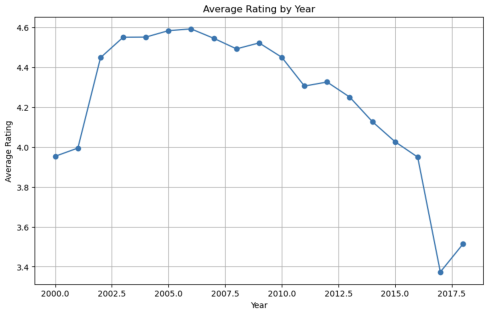

Figure 4: Average Ratings by Year

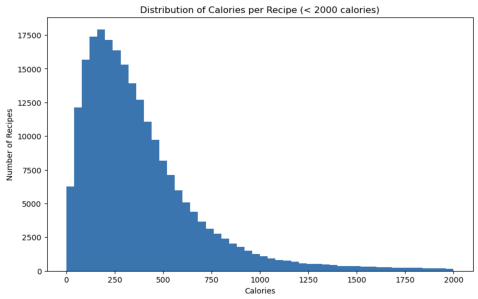

Figure 5: Distribution of Calories per Recipe

1.4 User and Recipe Engagement

1.4.1 User Engagement

Users submitted an average of 5 reviews each, with a median of 1 review per user, indicating that many users interact infrequently with the platform. The most active user submitted 7,671 reviews, highlighting the significant contribution of superusers.

1.4.2 Recipe Popularity

Recipes received an average of 4.89 reviews, with a median of 2 reviews per recipe. The most reviewed recipe garnered 1,613 reviews. The disparity between averages and medians suggests that a few recipes attract substantially more attention than others.

1.4.3 Review Text

Across all reviews there are 141322 unique words. The 5 most common words are ”I”, ”the”, ”and”, ”a”, and ”it”. The average length of a review is 51.8 words.

1.5 Key Findings

Several insights emerge from the analysis. High ratings dominate, indicating a bias toward positive feedback. Recipes are generally quick to prepare,

reflecting user preferences for efficiency. Simpler recipes with moderate ingredient counts are the most popular. Ratings have declined over time, which warrants further investigation. Finally, a small group of superusers contributes significantly to platform engagement, while most users are casual reviewers.

2 Predictive Task & Evaluation

Since our dataset contains ratings for each interaction between user and recipe, we have decided to predict a user's rating on a recipe. Predicting ratings relates directly to recommendation systems, by training a model that is able to accurately predict the ratings of recipes from users, the model can then be used to iteratively find the recipes that given users would rate the highest and thus be able to recommend those to a user.

Some baseline models for predicting ratings between user and recipe pairs are linear regression and basic latent factor models. The linear regression model will take some features of the interaction and use them to predict ratings. The latent factor model takes a user id, recipe id, and predict the rating from that information. This model will find the relationships between users and recipes, based solely upon interactions and no other data. This will serve as a good baseline since it is a model that creates bias terms for users and recipes, while also modeling user, recipe interaction through the inner product of γ_{user} and $\gamma_{recipes}$.

In order to test beyond the linear regression and latent factor model we will create other models that employ different methodologies to learn about the user's interactions with the recipes. Such models include a down-sampled latent factor model, a bag of words model, a deep learning model, and a deep learning sentiment analysis model.

To compare and assess our models' performance, we will use mean square error (MSE) metric, since it can informative for numeric prediction especially when predicting ratings. A low MSE would indicate that the predicted rating is close to the actual rating, and vice versa. Additionally, MSE is sensitive to large errors. This would easily help us ensure that our models are not predicting too far from the actual value.

Finally, to consider what features our models would take into account for, we conducted a random forest feature importance analysis on our datasets. The analysis gave us the most important features, they are: (1) recipe age, how old the recipe is the time the review was written; (2) nutritional values, this includes calories, sugar, sodium, protein, saturated fat, total fat and carbohydrates, in that order; (3) recipe complexity, this includes steps per ingredient, minutes, number of ingredients and number of steps; (4) tags features, this includes the keywords "Occasion", "Number-of-servings", "Easy", "Low-in-something", and "Cuisine"). With these features in mind, we will be able to incorporate them into the training of our models to accurately predict ratings.

For bag of words and deep learning sentiment analysis textual data cannot be directly fed into models and must be preprocessed in a way that the model can understand. For bag of words the review text must be converted into a one hot vector type of data, which represents the counts for each word in the text. This vector can then be used to produce rating predictions. Similarly the deep learning sentiment analysis model, requires that the reviews be tokenized. In order to preprocess and tokenize the reviews, we built a vocabulary based on the training data and this to tokenize each review.

3 Predictive Models

The six predictive models that we have explored are linear regression model, latent factor model, down-sampled latent factor model, deep learning model, bag of words model, and deep learning sentiment analysis model.

Model	MSE
Linear Regression	1.56
Latent Factor	1.75
Down-sampled Latent Factor	2.63
Deep Learning	1.48
Bag of Words	1.30
Deep Learning Sentiment Analysis	1.24

3.1 Baseline Linear Regression Model

The simplest model that we could utilize was a linear regression model. In this model, we used only one feature, which is the recipe age at the time of the review. The distribution of the results of this model mimics that of the dataset we have, with over 95.98% of predicted ratings to be at least a 4. The model resulted in a MSE of 1.5634. To further optimize the model, we added in the nutritional value of each recipe as features. This model only improved slightly, with a MSE of 1.5631 and

a skewed prediction distribution with 95.96% of ratings being a 4 or higher. We can use this model as a good baseline for our upcoming models, given that it is naively implemented.

3.2 Baseline Latent Factor Model

One of the primary rating prediction and recommendation models covered in class was latent factor due to its ability to model user and item biases as well as user-item interactions. Since our task is rating prediction we wanted to compare a basic latent factor model to other models. In order to build this basic latent factor model we used SVD from the Surprise library. The baseline Latent Factor Model was able to achieve a modest 1.75 MSE. However, upon further investigation we discovered that the model was disproportionately guessing 5 star ratings for all interactions below 5 stars. This makes sense since a disproportional amount (about 72%) of the data are 5 star reviews. In order to counter balance, one strategy that we deployed in the latent factor model was down-sampling 5 star ratings, therefore ensuring that there is a more equal distribution of rating data. While this did improve the MSE of ratings below 4 stars, as expected it increased the overall MSE. This is, therefore, an implementation and model deployment decision, whether it is more important to have a higher MSE at the expense of predicting higher average ratings, or it is more important that we predict low ratings, when the ratings are truly low, at the expense of lowering ratings and increasing MSE. This provided a comprehensive baseline model against which more complex models can be compared with.

3.3 Deep Learning Model

The deep learning model predicts recipe ratings by examining relationships among various recipe attributes and their influence on user satisfaction. It integrates key features, including the age of the recipe at the time of review, nutritional metrics such as calorie and protein content, measures of recipe complexity like the number of steps per ingredient and total cooking time, and categorical information derived from recipe tags.

The model is structured as a neural network with a four-layer architecture, progressively narrowing from 128 to 64, 32, and finally a single output node. Each layer employs ReLU activation functions to capture non-linear relationships, with dropout regularization rates of 0.3 and 0.2 applied between layers to mitigate overfitting. Batch normalization is utilized to stabilize the training process and accelerate convergence, enabling the network to identify subtle patterns in the data that simpler methods might overlook. By incorporating both nutritional and preparation-related features, the model learns complex interdependencies that contribute to user ratings.

During testing, the model achieved an MSE of 1.48, demonstrating its capability to provide accurate predictions. An advantage of this approach is its ability to estimate ratings for new recipes based solely on their attributes, without relying on user interaction data or textual reviews. This makes the model particularly valuable for predicting the potential success of newly introduced recipes, complementing traditional methods that depend on historical user feedback.

3.4 Sentiment Analysis

3.4.1 Bag of Words Model

Our bag of words approach to sentiment analysis was similar to that covered in class. Initially we separated our dataset into test and train datasets. On the train dataset, we extracted a dictionary of words and their counts after removing all punctuation. Then the dictionary was sorted to only include the 1000 words with the highest counts. Then for each review a one hot encoding was used for the number of times a word within the 1000 words occurred in the review. Finally, the training reviews and ratings were used to fit a linear ridge model. When tested on our test set we found a 1.42 MSE, which is a significant improvement upon the 1.75 MSE from the baseline latent factor model. This makes sense since the reviews have meaningful words which a model can learn to accurately predict ratings. For example the 3 words with the highest weight are excellent, outstanding, and fantastic, while the 3 words with the most negative weight are sorry, rate, and bland. This confirmed our hypothesis that using the reviews to predict ratings would result in lower MSE because there are more telling features.

3.4.1 Deep Learning Sentiment Analysis Model

This model used the text reviews, similar to bag of words, but leveraged deep learning in order to model more complex interactions and meanings between words. In order to prepare the data for this model, we created a vocabulary for our model

which was built by taking the training data and creating a dictionary for each unique word and a corresponding id number for each word. This allowed each of the reviews to be tokenized before being passed into the neural network. The deep learning model is two feed forward linear layers that with relu as the activation function. After experimenting with different hidden layer sizes, 128 was found to give the smallest MSE value. Similarly, the number of epochs was limited to 10 to avoid over fitting and the learning rate was set to 0.001. One of the unsuccessful attempts at creating this model was using an output with 6 different nodes to represent the different ratings, however, after comparison having a single output node was more accuracte, which is due to increased continuity, rather than having discrete ratings.

The deep learning model significantly outperforms the bag-of-words approach in terms of MSE, indicating that it is better able to analyze sentiment and model interactions between words. This makes sense since the embedding layer allows the model to learn nuanced relationships between words, while bag-of-words methods only count word occurrences without considering order or context. Another significant change is that this model tokenizes all words encountered in the training data, rather than limiting the vocabulary to the top 1000 most common words. This ensures richer input data, although unseen words are still represented using ¡unk¿. While this model outperforms all other models we tested, it is computationally and time expensive, in addition to being memory-intensive due to storing large vocabularies.

4 Literature Related to the Problem

4.1 Dataset Origin

This dataset was used in the paper, Generating Personalized Recipes from Historical User Preferences, which explores recipe generation. This paper describes the task of generating recipes based on user preferences when only given limited details about a wanted recipe, such as recipe name, key ingredients, and number of calories. This paper describes how the dataset was collected from 18 years worth of user recipe interaction from food.com.

Much of the literature that we explored used similar datasets, for example one used data from allrecipes.com. Another approach taken by Harvey, Morgan and Ludwig (2013) chose to take user data from their own surveys. Initially they collected 912 of the most popular recipes on a website and asked users to fill out a survey where they would rate a randomly selected recipe and optionally fill out an explanation for their rating.

4.2 Related Literature

The issue of recipe recommendation has been extensively explored, with researchers investigating various techniques to model user preferences, recipe attributes, and contextual factors.

Shah, Gaudani, and Balani (2016) focused on hybrid approaches combining content-based filtering with collaborative filtering to address sparsity challenges in culinary recommendation systems. They proposed two methods: one utilizing K-Nearest Neighbors with content and rating matrices to personalize recommendations based on recipe and user similarities, and another employing Stochastic Gradient Descent for model-based predictions, achieving reduced RMSE compared to traditional models. The study emphasized the importance of integrating diverse recipe features, such as preparation methods, cuisines, and dietary considerations, to enhance personalization.

Trattner and Elsweiler (2017) reviewed key challenges in food recommendation systems, including modeling complex user preferences, managing dietary constraints, and balancing taste with nutritional value. They discussed foundational methods like collaborative filtering, content-based filtering, and hybrid approaches while introducing health-aware recommender systems that incorporate nutritional metrics, such as calories and macronutrients, to promote healthier eating habits.

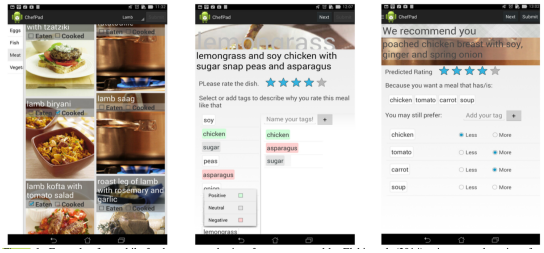

Figure 6: Mobile food recommender interface using ratings and tags. Adapted from Trattner and Elsweiler (2017).

Patil and Potdar (2019) examined various recipe recommendation techniques, including content-based and collaborative filtering approaches, while

addressing challenges like ingredient heterogeneity, cooking procedures, and data sparsity. They introduced a model for ingredient-based recommendations that constructs recipe vectors to improve user-recipe recommendation. Their study proposed integrating health conditions and user preferences through a dietary machine learning recommendation system.

Tian (2022) proposed a Hierarchical Graph Attention Network (HGAT) designed to capture multi-relational data, such as user-recipe-ingredient interactions. This model effectively integrates heterogeneous data like textual descriptions and user preferences to improve recommendation accuracy. Their findings highlighted the effectiveness of graph-based approaches in enhancing personalized recipe recommendations.

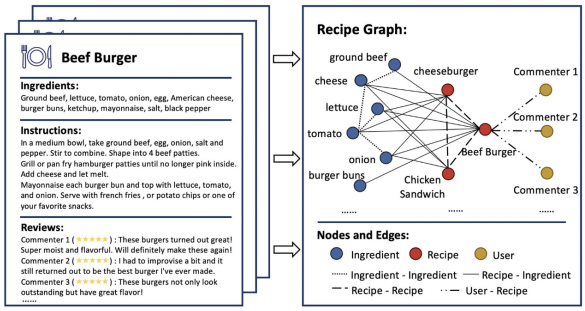

Figure 7: Visualization of the recipe graph with nodes (users, recipes, and ingredients) and their relations. Adapted from Tian (2022).

Other models and interesting areas of exploration mentioned in research are multimodal approaches and ones that emphasize health awareness and explainable recommendation. The multimodal approaches use additional information such as text and images of recipes to aid in suggestions. While health awareness and explainable recommendation focus on creating rational for recommendations.

4.3 State-of-the-Art Methods

Current techniques for recipe recommendation include collaborative filtering, content-based filtering, graph-based models, multimodal approaches, and hybrid methods. Collaborative filtering methods, like matrix factorization (e.g., Singular Value Decomposition), are effective for personalization but face challenges with data sparsity. Content-based filtering, using attributes like ingredients and cooking steps, is enhanced by text-based approaches such as TF-IDF and word embeddings. Graph-based models, including Graph Neural Networks and hierarchical attention models like Tian's HGAT, effectively capture relational complexities in user-recipe-ingredient data.

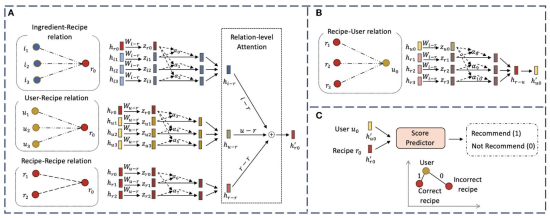

Figure 8: Architecture of the Hierarchical Graph Attention Network (HGAT) for recipe recommendation. Adapted from Tian (2022).

Multimodal approaches combine visual, textual, and structured data for richer recipe representations, while hybrid methods address limitations like cold-start problems by integrating multiple techniques.

4.4 Comparison of Findings

Our findings align with existing research in several ways. For example, our implementation of Singular Value Decomposition mirrors the performance reported by Trattner and Elsweiler (2017). Additionally, the incorporation of user-recipe-ingredient relationships in our deep learning model are similar to the graph based approaches, explored in Tian (2022).

Our analysis differs from much of the literature since our goal is to accurately predict ratings, while many papers focused on optimizing health and generating new recipes. In addition to looking at recipe and user features we also explored prediction capabilities solely based on review data. However, we encountered similar issue with data sparsity for users and recipes which can impact prediction abilities. We also struggled with data that was highly skewed toward 5 star ratings.

4.5 Conclusions from Literature

The reviewed literature underscores the importance of combining diverse data types, such as user interactions, recipe attributes, and nutritional metrics, to enhance recommendations. Our exploration aligns with existing research in emphasizing personalization and user preferences, while also highlights unique challenges, including rating biases and data sparsity.

5 Conclusion and results

Our analysis of the Food.com dataset yielded several key insights into predicting recipe ratings through various modeling approaches. The models we developed demonstrated a clear progression in predictive accuracy, with our deep learning sentiment analysis model achieving the best performance (MSE = 1.24), followed by bag of words (MSE = 1.30), deep learning model (MSE = 1.48), linear regression (MSE = 1.56), and latent factor models (MSE = 1.75).

A crucial finding was the superiority of text-based features in rating prediction compared to traditional collaborative filtering methods. The success of our sentiment analysis models suggests that review text can capture user satisfaction more accurately than certain features of recipes or users. This was particularly evident in our deep learning sentiment analysis model, which effectively learned complex word interactions and relationships.

For non text based model approaches, we did find that recipe metadata (ie. cooking time, ingredient count, nutritional information) contributed to increased prediction accuracy, indicating that practical aspects of recipe preparation influence user ratings alongside taste preferences.

Several challenges emerged during our analysis. The computational demands of processing large text datasets required careful optimization, particularly for our deep learning models. Additionally, the heavily skewed rating distribution (predominantly 5-star ratings) posed difficulties in model training and evaluation.

For our main deep learning models the model parameters are more obscure than linear regression and bag of words which have weights for certain features. The parameters of this model are obscured within the weights and biases of the neural network who's multiple layers are able to model complex interactions and nuances of the data.

The feature representations that worked well were text data of the reviews and specific recipe data used in our deep learning model (age of the recipe at the time of review, nutritional information, etc.). Conversely, using solely user-recipe relationships with the latent factor model resulted in diminished accuracy. Similarly, using a single feature in our linear regression model also resulted in worse accuracy. This makes sense because our latent factor and linear regression models do not take advantage of the additional data available for analysis. Therefore, models that use multiple meaningful data points or contextually rich data (recipe reviews) will outperform models with single or few features.

Our overall findings show the effectiveness of text-based models for predicting recipe ratings. The deep learning sentiment analysis model achieved the best performance out of all our models, significantly outperforming our other traditional models. This performance gap suggests that review text contains richer predictive information than numerical features alone. This is further proved with the relative success of bag of words compared to collaborative filtering. We also found that nutritional information and recipe complexity provided meaningful signals, though less impactful than text features.

Acknowledgments

ChaptGPT was used to aid with proofreading the paper and debugging code for models.

References

Harvey, M., Ludwig, B., and Elsweiler, D. (2013). You are what you eat: Learning user tastes for rating prediction.

Majumder, B. P., Li, S., Ni, J., and McAuley, J. (2019). Generating personalized recipes from historical user preferences.

Patil, S. T. and Potdar, R. D. (2019). Recipe recommendation systems: A review.

Tian, Y., Zhang, C., Metoyer, R., and Chawla, N. V. (2022). Recipe recommendation with hierarchical graph attention network.

Trattner, C. and Elsweiler, D. (2017). Food recommender systems: Important contributions, challenges and future research directions.

(Trattner and Elsweiler, 2017) (Tian et al., 2022) (Patil and Potdar, 2019) (Majumder et al., 2019) (Harvey et al., 2013)