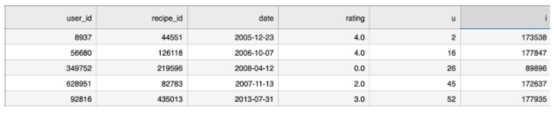
Predictive System for Forecasting Recipe Ratings Using User Interaction Data

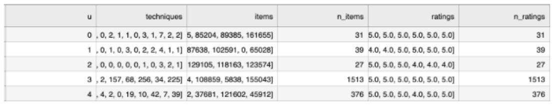
I. Dataset

Our dataset from Food.com contains more than 230k different recipes with reviews from over 225k users that total to 1.1M+ reviews (interactions) from the time of 2000-2018 [1]. We worked with two files, RAW_interactions.csv and RAW_recipes.csv. RAW_interactions holds the information about each review, major features being the user's id, recipe id and a rating on a scale of 0-5 stars.



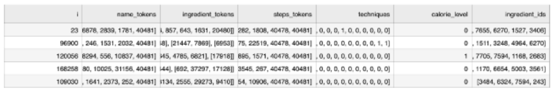
user_id	recipe_id	date	rating	u	i
8937	44551	2005-12-23	4.0	2	173538
56680	126118	2006-10-07	4.0	16	177847
349752	219596	2008-04-12	0.0	26	89896
628951	82783	2007-11-13	2.0	45	172637
92816	435013	2013-07-31	3.0	52	177935

Table 1: Example of the first few rows of interactions_test.csv, interactions_train.csv, and interactions_validation.csv.



u	techniques	items	n_items	ratings	n_ratings
0	, 0, 2, 1, 1, 0, 3, 1, 7, 2, 2]	5, 85204, 89385, 161655]	31	5.0, 5.0, 5.0, 5.0, 5.0, 5.0]	31
1	, 0, 1, 0, 3, 0, 2, 2, 4, 1, 1]	87638, 102591, 0, 65028]	39	4.0, 4.0, 5.0, 5.0, 5.0, 5.0]	39
2	, 0, 0, 0, 0, 0, 1, 0, 3, 2, 1]	129105, 118163, 123574]	27	5.0, 5.0, 5.0, 4.0, 4.0, 4.0]	27
3	, 2, 157, 68, 256, 34, 225]	4, 108859, 5838, 155043]	1513	5.0, 5.0, 5.0, 5.0, 5.0, 5.0]	1513
4	, 4, 2, 0, 19, 10, 42, 7, 39]	2, 37681, 121602, 45912]	376	5.0, 5.0, 5.0, 4.0, 5.0, 5.0]	376

Table 2: These values are the first few rows of PP_users.csv, where we can see the ratings as well as how many users have rated this particular recipe before and a list of those ratings.



i	name_tokens	ingredient_tokens	steps_tokens	techniques	calorie_level	ingredient_ids
23	6878, 2839, 1781, 40481]	), 857, 643, 1631, 20480]]	282, 1808, 40478, 40481]	, 0, 0, 0, 1, 0, 0, 0, 0, 0]	0	, 7655, 6270, 1527, 3406]
96900	, 246, 1531, 2032, 40481]	48], [21447, 7869], [6953]]	75, 22519, 40478, 40481]	, 0, 0, 0, 0, 0, 0, 0, 1, 1]	0	, 1511, 3248, 4964, 6270]
120056	8294, 556, 10837, 40481]	345, 4785, 6821], [17918]]	895, 1571, 40478, 40481]	, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1	, 7705, 7594, 1168, 2683]
168258	80, 10025, 31156, 40481]	44], [692, 37297, 17128]]	3545, 267, 40478, 40481]	, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0	, 1170, 6654, 5003, 3561]
109030	, 1641, 2373, 252, 40481]	3134, 2555, 29273, 9410]]	54, 10906, 40478, 40481]	, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0	[3484, 6324, 7594, 243]

Table 3: This table comes from PP_recipes.csv, and it holds recipe name, ingredient, and step tokens as well as calorie levels and ingredient ids.

An interesting finding was that 5-star ratings were the most frequent, followed by a decreasing trend from 4 to 1 stars, while 0-star ratings also had a relatively high count.

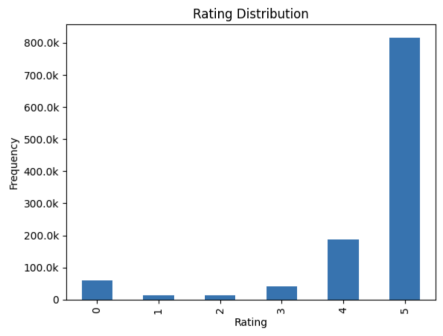


Figure 1: Most of these recipes are rated with 5 stars.

The data is imbalanced, with a skew toward higher star ratings, making it challenging for the model to predict lower ratings accurately. The dataset also includes written reviews, which can be analyzed using TF-IDF for sentiment analysis to improve predictions. However, about 169 reviews lack text, making them unusable for a text-based model.

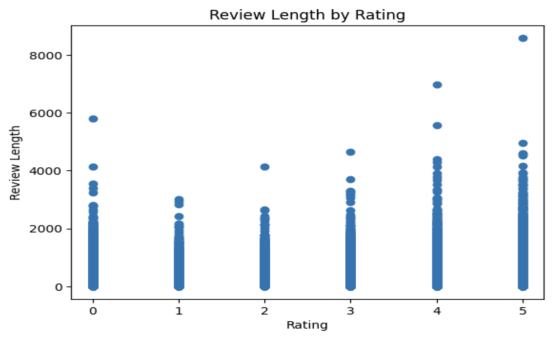


Figure 2: Plotting the relationship between the review length and rating.

While looking at the reviews, we were curious to see if there was some correlation between the length of a review (in characters) to the actual ratings, but we can see above that there is no relationship between the two as we see similar lengths of ratings across the board. With an average length of about 278.40 characters per review, we see that they all tend to be on the smaller side regardless of rating.

The RAW_recipes file holds the information for each recipe such as the number of steps, ingredients, and minutes to prepare.When looking through our data, there are 2 entries that hold the most minutes, one with 2,147,483,647 minutes or 4083 years, this entry is clearly input incorrectly. The next biggest one is 1,051,200 minutes or 730 days which is named "How to preserve a husband", clearly a joke, which still exists on the Food.com website to this day. Since they are both wrongly input or jokes, they will be removed to help keep the integrity of the data true.

II. Predictive Task

Model Evaluation

To evaluate model performance, we used three key metrics:

1. **Mean Absolute Error (MAE):** MAE measures the average error between predicted and actual ratings, where lower values indicate higher accuracy.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|pred_i - actual_i\right|$$

2. **Percentage of True Predictions (PTP):** Represents the proportion of predictions where the rounded rating matches the actual rating, providing a simple measure of accuracy.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

3. **Prediction Error Distance:** Quantifies the spread of errors between predicted and actual ratings, offering insights into prediction variance.

Baseline Models

We used two baseline approaches to compare model performance and establish robustness:

Baseline Model 1: Item Average Ratings

Method: Predicts ratings using the average rating for each recipe (item) based on the training data. If a recipe's rating is unavailable, it falls back to the global average rating, or mean of all ratings in the training dataset. Predictions were then rounded to the nearest integer and clipped to ensure they fall within the valid range of ratings (1 to 5).

Baseline Model 2: Matrix factorization using Singular Value Decomposition (SVD)

Method: Implements a matrix factorization approach using the SVD algorithm to predict ratings based on user and item latent factors.

- **SVD Algorithm in Recommendations:** Singular Value Decomposition (SVD) is a mathematical technique that factorizes a matrix into three smaller matrices: U, Σ, and V^T. In the context of predicting ratings, the user-item interaction matrix (e.g., a matrix of users and their ratings for recipes) is decomposed such that U represents user preferences, V^T represents item characteristics, and Σ contains the singular values that scale the contributions of these factors.
- **Latent Factors**: Latent factors represent underlying features that influence user preferences and item characteristics that enable personalization and uncover hidden relationships.

Model Validity

We began with the careful division of the dataset into training and testing sets using an 80/20 split. This split ensures that the model is able to learn underlying patterns and relationships, while having an independent set for evaluating how well the model generalizes to unseen data.

The validity of the model's performance was then determined by several key factors:

1. **Generalization:** Ensuring the model's ability to perform well on the test set demonstrates its capacity to generalize from the training data to new, unseen reviews from users.
2. **Evaluation Metrics:** Robust evaluation metrics like MAE and percentage of true predictions provide objective measures of accuracy and allow for comparisons between different models.
3. **Reproducibility and Consistency**: By using the 80/20 split consistently across all models, I ensured that each model is evaluated under the same conditions. Consistent validation results across multiple runs further support the reliability and validity of the model.

Features Used

To train the NLP regression model, we utilized both text-based features extracted from user reviews and a derived sentiment score. Below is a detailed description of the features and the processing steps involved:

Features 1: TF-IDF (Term Frequency-Inverse Document Frequency)

Purpose: Captures the importance of a set of words and phrases in the review text, weighted by their frequency in the dataset and uniqueness across reviews.

Text Preprocessing: First stop words (common words like "the", "and") were removed, and then bigrams (pairs of consecutive words) were included to capture contextual information.

Feature Engineering: The top 10,000 most relevant features were selected based on frequency, with a minimum document frequency (min_df) of 5 to eliminate rare words and a maximum document frequency (max_df) of 0.85 to exclude overly common terms.

Output: Sparse matrices for training and validation containing numerical representations of the reviews.

Feature 2: Sentiment Polarity

Purpose: Quantify the emotional tone of the review, with scores ranging from -1 (negative) to +1 (positive).

Processing: Sentiment polarity was calculated using the TextBlob library for each review. Missing reviews were filled with an empty string, resulting in a neutral sentiment score of 0.

Feature Representation: Sentiment scores were reshaped and concatenated with TF-IDF features to enhance the model with an emotional aspect of the text.

Data Processing

1. **Data Cleaning & Preparation:** Merged interaction and recipe datasets on recipe_id to include both user-generated ratings and reviews, along with recipe details. Handled missing reviews by filling with empty strings.
2. **Sentiment Analysis:** Created a new column in the training and validation datasets for the calculated sentiment polarity. This step added a continuous numerical feature representing the review's tone.
3. **Text Vectorization with TF-IDF:** Tokenized the reviews and transformed them into a matrix of TF-IDF features, where each row corresponds to a review, and each column represents words or bigrams. Fitted the vectorizer on the training dataset and transformed both training and validation datasets to ensure consistent feature space.
4. **Feature Combination:** Combined the sparse matrix of TF-IDF features with the dense sentiment polarity feature using scipy.hstack. This resulted in a single feature set containing both text and sentiment information.
5. **Model Training and Validation:** Trained a linear regression model on the combined feature set and validated its predictions using evaluation metrics.

III. Our Model

Model Description

Our model is a linear regression-based natural language processing (NLP) system that predicts recipe ratings by combining text analysis and sentiment evaluation. It leverages TF-IDF (Term Frequency-Inverse Document Frequency) to extract textual features from user reviews, capturing the importance of unique words and phrases, while incorporating sentiment polarity scores derived from TextBlob to quantify the emotional tone of each review. The features are combined into a sparse matrix and used to train the model on user ratings, ensuring a robust representation of both textual content and sentiment. The model is evaluated using RMSE, MAE, and accuracy within a ±0.5 range of actual ratings, offering insights into its predictive performance.

Model Justification

The decision to use a linear regression model combined with TF-IDF features and sentiment polarity was driven by the nature of the problem and the structure of the data. Predicting recipe ratings based on textual reviews is a regression task where the numerical rating depends on the review's content and tone. TF-IDF captures the importance of unique words and phrases in the text, effectively encoding the semantic and contextual aspects of the reviews. Adding sentiment polarity as a feature provides an emotional dimension to the predictions, allowing the model to consider the positive or negative tone of reviews. Together, these features offer a rich representation of the data, enabling the model to generalize effectively.

Linear regression was chosen for its simplicity, interpretability for predicting numerical ratings. This approach allows the model to efficiently map the combination of textual features and sentiment scores to a continuous rating scale. It balances complexity and performance.

Optimization

The model was optimized by carefully selecting and preprocessing features to ensure relevance and efficiency. First, TF-IDF vectorization was fine-tuned with a maximum of 10,000 features, inclusion of bigrams, and frequency thresholds (min_df=5, max_df=0.85) to capture meaningful patterns while avoiding noise. Sentiment polarity, calculated using TextBlob, was added to provide an emotional dimension to the predictions. These features were combined into a single feature set to balance textual content and sentiment insights.

The dataset was split into training and validation sets (80/20) to evaluate generalization and avoid overfitting. Hyperparameters such as the number of TF-IDF features and n-gram range were adjusted iteratively based on performance metrics like RMSE and MAE. Predictions were clipped to the valid range (1 to 5) to ensure practical ratings, and metrics were recalculated to refine performance. This process ensured the model was both robust and efficient for the given task.

Scalability/Overfitting

Scalability challenges arose due to the high dimensionality and memory consumption of TF-IDF features, with 10,000 features and bigrams. To address this, sparse matrices were utilized to optimize memory usage and fine-tuned TF-IDF parameters (e.g., min_df and max_df) to exclude irrelevant features.

Overfitting risks were mitigated by using an 80/20 train-validation split to evaluate generalization and monitoring metrics like RMSE and MAE on unseen data. The inclusion of generalizable features like sentiment polarity and the simplicity of linear regression helped prevent over-reliance on training data, ensuring a balance between performance and robustness.

Other models for comparison

Model 1: Neural Network Latent-Based

This model uses embeddings to represent users and items (recipes) in a lower-dimensional "latent space," where their relationships can be learned. These embeddings study and learn hidden patterns or preferences, which the model combines with dense layers to predict ratings.

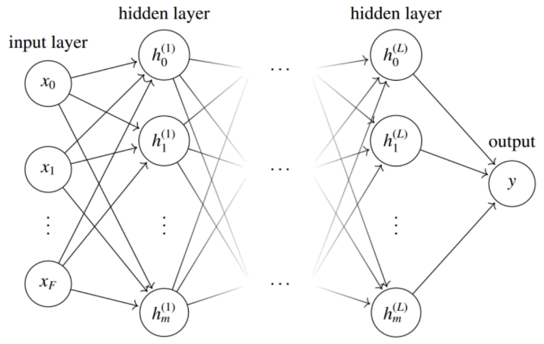


Figure 1: This diagram from *Neural Collaborative Filtering (He et al. 2017)* is an example of how there is a multilayer perceptron to understanding and studying the relationship between gamma_u and gamma_i.

By mapping each user and recipe to dense embedding vectors, the model captures latent features that reflect complex relationships between users and recipes. The embeddings are concatenated and passed through fully connected layers with ReLU activation and dropout regularization, enabling the model to learn nonlinear patterns while reducing the risk of overfitting. The output layer uses linear activation, making it suitable for regression tasks like predicting ratings. The model is compiled with the Adam optimizer and mean squared error (MSE) as the loss function, focusing on accurate predictions. Training incorporates early stopping to prevent overfitting, halting if the validation loss does not improve after three epochs. This approach is ideal for personalized recommendation systems, as it effectively models latent, nonlinear interactions between users and items while generalizing well to unseen data.

Model 2: Jaccard similarity

Calculates Jaccard similarity between users to help us predict the rating of a recipe. This model takes into account what other users that rated the same recipe rated things to help determine similar ratings. Similarity (Equation 1) is heavily reliant on interactions between users and recipes to be very interconnected.

$$\mathrm{Sim}(u,v) = \frac{\sum_{i \in \overset{\text{items rated by both users}}{I_u \cap I_v}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \overset{\text{average rating by user } v}{\bar{R}_v})}{\sqrt{\sum_{i \in I_u \cap I_v} (R_{u,i} - \bar{R}_u)^2 \sum_{i \in I_u \cap I_v} (R_{v,i} - \bar{R}_v)^2}}$$

Equation 1: This equation is suitable for numerical ratings and essentially subtracts the average minus the values to conclude if the predictions are below-average (negative) or above-average (positive).

This measurement was instrumental in identifying patterns of shared preferences and interactions within the dataset. By calculating the similarity between users, we were able to identify clusters of individuals with comparable tastes, which helped inform recommendations for recipes that one user enjoyed but the other had not yet tried. Similarly, measuring the overlap between recipes helped us group those with shared audiences, facilitating the identification of recipes likely to appeal to similar users.

Unsuccessful attempts

The Neural Network model was unsuccessful in the sense that the complexity of the model resulted in a very long runtime. Also, this model was extremely susceptible to overfitting very early on in training the model. Otherwise, we trained a fairly successful model with findings that were helpful and relevant to our task but not our best solution.

Our Jaccard similarity model was not accurate enough to be considered among our choices as it performed worse than our baseline of always predicting mean in terms of MSE and RMSE. With it being so lacking in accuracy, there was no way that we would consider it to be our final model.

Strengths and weaknesses of other models

The Neural Network model's key strength is its ability to learn latent embeddings for users and recipes, capturing hidden patterns to provide personalized recommendations. Its layered design models complex, nonlinear relationships, enhancing predictive accuracy, while dropout layers and early stopping reduce overfitting. The model has weaknesses, including early susceptibility to overfitting, as evidenced by increasing validation loss despite decreasing training loss, which was addressed with early stopping (Table 4). It requires substantial interaction data, posing challenges in sparse scenarios or with new users/recipes. Additionally, training is computationally expensive and time-intensive, given the dataset size of 231,637 recipes and 226,570 users.

Epoch	Training Loss (MSE)	Validation Loss (MSE)	Training Time
1	1.4916	1.6215	1205
2	0.8683	1.5561	1182
3	0.7115	1.6694	1265
4	0.6338	1.8539	1198

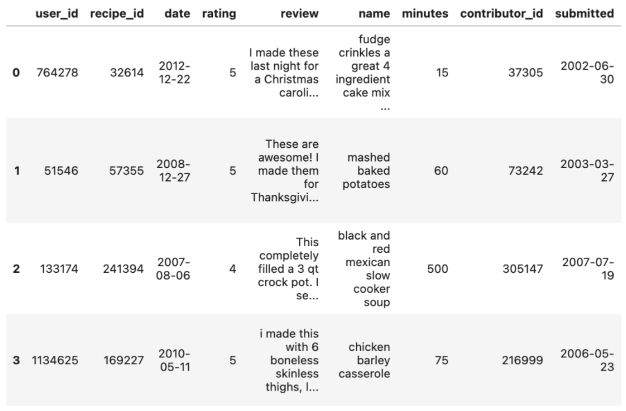
Table 4: A result of our findings from the Neural Network Latent Based model.

For our second comparison, a Jaccard similarity model is fairly simple to set up and it is able to work really well as a recommender system, but is able to be transformed to predict ratings. This model can pick up on things that other models don't look for such as ratings from others and weighing them depending on how related they are. The downsides to this model is that it is heavily reliant on a dataset that is interconnected with itself to produce these predictions, meaning that it might be good on a website like amazon where people are buying many items from the same website.

IV. Literature

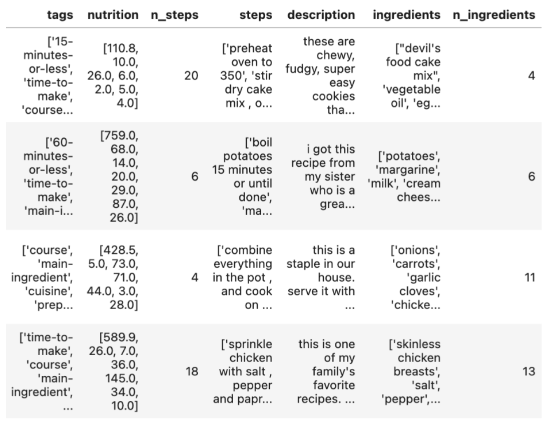
Dataset

The dataset from Food.com includes (1) ratings and reviews, (2) recipe details like name, description, ingredients, and directions, (3) recipe categories, and (4) nutrition information. For example, it contains columns for cook time, nutrition, steps, tags, ratings, and reviews.



	user_id	recipe_id	date	rating	review	name	minutes	contributor_id	submitted
0	764278	32614	2012-12-22	5	I made these last night for a Christmas caroli...	fudge crinkles a great 4 ingredient cake mix ...	15	37305	2002-06-30
1	51546	57355	2008-12-27	5	These are awesome! I made them for Thanksgivi...	mashed baked potatoes	60	73242	2003-03-27
2	133174	241394	2007-08-06	4	This completely filled a 3 qt crock pot. I se...	black and red mexican slow cooker soup	500	305147	2007-07-19
3	1134625	169227	2010-05-11	5	i made this with 6 boneless skinless thighs, I...	chicken barley casserole	75	216999	2006-05-23

Figure 2: The first half of the combined dataframe that we prepared for proper use.



tags	nutrition	n_steps	steps	description	ingredients	n_ingredients
['15-minutes-or-less', 'time-to-make', 'course...	[110.8, 10.0, 26.0, 6.0, 2.0, 5.0, 4.0]	20	['preheat oven to 350', 'stir dry cake mix , o...	these are chewy, fudgy, super easy cookies tha...	["devil's food cake mix", 'vegetable oil', 'eg...	4
['60-minutes-or-less', 'time-to-make', 'main-i...	[759.0, 68.0, 14.0, 20.0, 29.0, 87.0, 26.0]	6	['boil potatoes 15 minutes or until done', 'ma...	i got this recipe from my sister who is a grea...	['potatoes', 'margarine', 'milk', 'cream chees...	6
['course', 'main-ingredient', 'cuisine', 'prep...	[428.5, 5.0, 73.0, 71.0, 44.0, 3.0, 28.0]	4	['combine everything in the pot , and cook on ...	this is a staple in our house. serve it with ...	['onions', 'carrots', 'garlic cloves', 'chicke...	11
['time-to-make', 'course', 'main-ingredient', ...	[589.9, 26.0, 7.0, 36.0, 145.0, 34.0, 10.0]	18	['sprinkle chicken with salt , pepper and papr...	this is one of my family's favorite recipes. ...	['skinless chicken breasts', 'salt', 'pepper',...	13

Figure 3: The second half of the combined dataframe that we prepared for proper use.

To prepare this data, we merged two separate DataFrames, interactions_df and recipes_df, using the recipe_id as the common key. The interactions_df contains user interactions with recipes (e.g., ratings, reviews), while recipes_df includes detailed recipe information. To prepare for the merge, the recipes_df column id is renamed to recipe_id. After merging, the combined dataset is shuffled to ensure randomness, with a consistent shuffle achieved using a fixed random seed (random_state=42). The resulting combined_df contains both user interaction data and recipe details, structured for further analysis or modeling.

Similar datasets and methods of study

There are other similar datasets that contain food information just like Food.com, and there are also datasets that also study user and item relationships like Goodreads (book reviews and ratings), Amazon Reviews (product reviews of items), IMBd or Rotten Tomatoes (movie reviews and ratings), etc. All of these datasets share characteristics with the dataset that we studied, which includes user-item interactions and numerical ratings for studying personalization and recommendation systems.

State-of-the-art methods for analyzing this type of data include matrix factorization or deep learning techniques like transformers. These models are intended to capture and study intricate and complex patterns in the data, which include nonlinear relationships and dependencies. Other methods include neural networks or other reinforcement learning that optimize long-term user satisfaction by modeling sequential interactions.

Comparison of conclusions between works

Our conclusions were somewhat similar to other work that also aimed to predict ratings by using a model focused on word analysis with TF-IDF using a linear model as our way of predicting the rating. Other than that we both ended up with similar models that utilized TF-IDF alongside a linear regression model to predict ratings. They also found models that focused on textual analysis to be the most effective model among the ones they tested, but determined that it was impractical since there are times that people don't leave any written review which would cause problems for text based models [2]. Another group that also utilized a model using TF-IDF came to the conclusion that there was still much more work to be done in terms of the Natural Language part of their project and to improve the way they do sentiment analysis[3].

V. Conclusion and Findings

Results and Conclusions

Final Model: NLP & TF-IDF + Sentiment

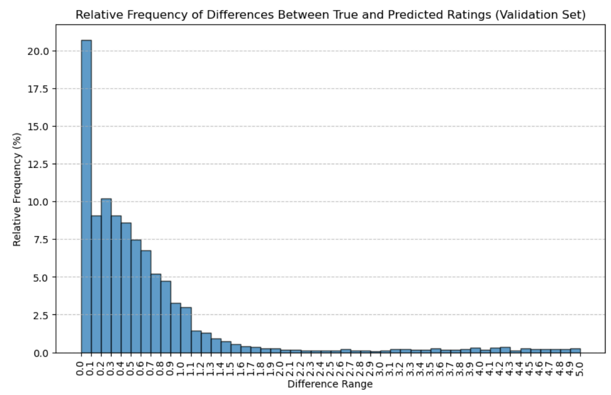


Figure 4: The figure illustrates the NLP model's strong accuracy, with most prediction errors near zero and larger deviations being rare.

The NLP model achieved a **MAE of 0.6379**, indicating a high level of precision, with predictions deviating from true ratings by an average of 0.64 points. This is a significant improvement over the 2 baseline models, demonstrating the effectiveness of combining textual features and sentiment analysis. A **PTP of 57.61%** further highlights the model's ability to generate predictions that closely align with actual user ratings, with over half of its predictions falling within ±0.5 of the true value.

Figure 4 underscores this performance, showing a concentration of errors near zero and a steep decline as errors increase. This clustering indicates the model's robustness in capturing the nuanced relationship between review content, sentiment, and ratings. By leveraging TF-IDF to extract meaningful textual patterns and incorporating sentiment polarity to account for emotional tone, the model achieves a comprehensive representation of user feedback, enabling accurate and personalized predictions. These results validate the model's design as both effective and practical for real-world recommendation systems.

Baseline Model 1: Item Average Ratings

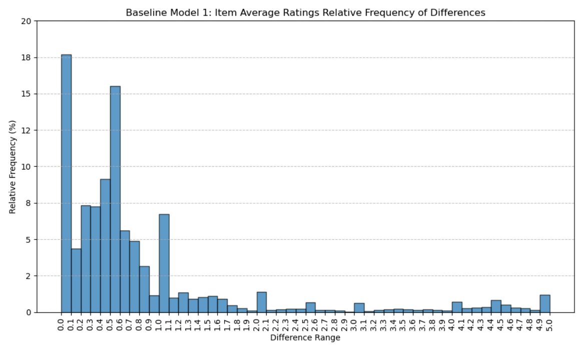


Figure 5: Error distribution for Baseline Model 1

Baseline Model 1 achieved a MAE of 0.8180 and a PTP of 46.70%. Compared to the NLP model, Baseline Model 1 performed worse, with higher errors (MAE: 0.8180 vs. 0.6379) and fewer predictions within ±0.5 of actual ratings (PTP: 46.70% vs. 57.61%), highlighting its inability to capture review nuances. The figure shows a broader distribution of errors, with a higher proportion of large deviations, reflecting the model's inability to effectively account for user-specific variability or nuanced feedback.

Baseline Model 2: SVD Matrix factorization

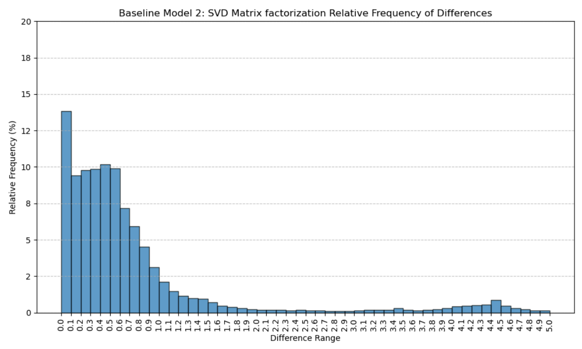


Figure 6: Error distribution for Baseline Model 2

Baseline Model 2, achieved a MAE of 0.7481 and PTP of 54.96%, showing improved accuracy over Baseline Model 1 but underperforming the NLP model (MAE: 0.6379, PTP: 57.61%). The error distribution was more concentrated near zero, indicating better performance in capturing user-item interactions compared to Baseline Model 1. However, it still displayed notable variability, suggesting room for improvement in handling complex, unstructured data.

Significance of results compared to others

The NLP model demonstrates clear advantages over both Baseline Model 1 (Item Average Ratings) and Baseline Model 2 (SVD Matrix Factorization), emphasizing the importance of integrating textual analysis and sentiment evaluation in recommendation systems. By achieving a **MAE of 0.6379** and a **PTP of 57.61%,** the NLP model outperforms both baselines, particularly excelling in accuracy and the ability to deliver predictions closely aligned with actual ratings. This performance underscores its effectiveness in capturing user sentiments and textual nuances, which are absent in traditional collaborative filtering approaches.

The significance of these results lies in the model's capacity to provide personalized predictions by leveraging unstructured review data and emotional tone. Unlike SVD, which focuses on latent user-item interactions, the NLP model bridges critical gaps by incorporating rich contextual information from reviews. The tighter clustering of prediction errors around zero and the steep decline in larger deviations further illustrate the model's ability to minimize significant mispredictions, making it particularly effective for datasets with extensive review content, such as Food.com.

These findings validate the NLP model as a robust and practical approach for enhancing recommendation systems, particularly in domains where user reviews are central to decision-making. Its superior performance highlights the need for models that go beyond numerical interactions and account for the textual and emotional dimensions of user feedback. By addressing limitations in simpler implemented baselines, the model with TF-IDF and Sentiment analysis sets a strong foundation for building systems capable of delivering accurate, user-specific recommendations in complex, real-world scenarios.

Feature representations

Effective Feature Representations

TF-IDF Features: We captured meaningful patterns in review text by emphasizing unique words and bigrams while filtering out noise with min_df and max_df thresholds. Resulted in strong predictive performance, as shown by the model's low MAE and high PTP, indicating its ability to leverage textual content effectively.

Sentiment Polarity: Provided an additional dimension by quantifying the emotional tone of reviews. Improved the model's ability to account for subjective user feedback, complementing the TF-IDF features.

Ineffective Feature Representations

Non-Textual Features : Recipe metadata (e.g., minutes, n_steps, n_ingredients) were excluded, as they did not directly capture the relationship between user reviews and ratings, limiting their relevance to the predictive task.

Sparse or Missing Reviews: Reviews with insufficient text or missing data reduced the effectiveness of both TF-IDF and sentiment features, limiting the model's ability to predict accurately for such cases.

Parameter Interpretation

The model's parameters represent the weights assigned to each feature, including the TF-IDF scores of review text and the sentiment polarity. These weights indicate the relative influence of specific words, phrases, and emotional tone on the predicted rating. Higher weights correspond to features with a stronger correlation to user ratings, while lower weights suggest less impact. The linear regression model assumes an additive relationship, meaning the final rating prediction is a weighted sum of these feature contributions, adjusted by the model's intercept. This parameter interpretation highlights the model's ability to quantify the importance of both textual content and sentiment in predicting user ratings.

Model Success and Failures

The proposed NLP model succeeded by effectively addressing the unique characteristics and challenges of the dataset while balancing complexity, accuracy, and practicality. It leveraged both structured and unstructured data—specifically, user review text and sentiment polarity—to predict ratings with precision. Achieving a MAE of 0.6379 and a PTP of 57.61%, the NLP model demonstrated its ability to extract meaningful patterns from the review content. TF-IDF captured the significance of individual words and bigrams, emphasizing context and relevance, while sentiment polarity quantified emotional tone, providing an additional dimension of user feedback. This integration allowed the model to generalize well across the dataset, even in cases where user-recipe interactions were sparse.

In contrast, the Jaccard similarity model failed due to the dataset's inherent sparsity. Many recipes had only one reviewer, and most users reviewed only a handful of recipes. These limited connections between users and recipes severely constrained the model's ability to establish meaningful relationships, as it relied entirely on shared preferences to make predictions. The absence of robust connections rendered the Jaccard model unreliable, with low accuracy and poor scalability for datasets lacking dense interaction data. Its failure highlights the importance of incorporating alternative data sources, such as text reviews, to overcome the limitations posed by sparse interactions.

The Neural Network model, while theoretically capable of capturing nonlinear relationships and hidden patterns, suffered from overfitting and impracticality in this context. Its complexity, combined with the dataset's moderate size and variability, caused it to overfit early, as evidenced by validation loss increasing while training loss continued to decrease. Additionally, the model's training time of over two hours made it infeasible to optimize within the project's constraints. Although its accuracy was promising, the trade-offs in efficiency and usability overshadowed its benefits, making it unsuitable for this application. This underscores that even sophisticated models can fail when their complexity is misaligned with the dataset and project goals.

The success of the NLP model reflects its alignment with the dataset's strengths and limitations. Unlike Jaccard or Neural Network models, the NLP approach directly utilized the universally available review text, bypassing issues of sparse interactions while maintaining interpretability and efficiency. Its relatively low computational requirements and ability to capture textual nuances made it a practical and effective solution, demonstrating that predictive success depends not only on model complexity but on its ability to exploit the most relevant data features for the task.

References

[1] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating Personalized Recipes from Historical User Preferences. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019), 5975-5981. DOI:https://doi.org/10.18653/v1/d19-1613

[2]Mayank Jain , Jonathan Woenardi , and Jash Makhija.Retrieved December 3, 2024 from https://jainmayank.me/reports/ratings-prediction.pdf

[3]Ning Yu, Desislava Zhekova, Can Liu, and Sandra Kubler. 2013. Do Good Recipes Need Butter? Predicting User Ratings of Online Recipes. (2013). Retrieved December 3, 2024 from https://cl.indiana.edu/~skuebler/papers/epi13.pdf