CSE 158/258 - Recommender Sys & Web Mining - FA24

Assignment 2 Report

Andrew Choi (A69033628), Chanbin Na (A18087468), Jonghee Chun (A69033997), Kenny Hwang (A99021639)
[Github Repo]

Abstract

This report analyzes the Food.com dataset to predict recipe ratings from review texts using linear regression with ridge regularization and to classify user preferences (ratings ≥ 4) using a Random Forest classifier. The rating prediction model achieved an MSE of 1.2823, an MAE of 0.7039, and an R² score of 0.1793, while also identifying the positivity of words in the reviews. The preference prediction model achieved an F1 score of 0.96 and highlighted the importance of personalized preferences.

1. Exploratory Dataset Analysis

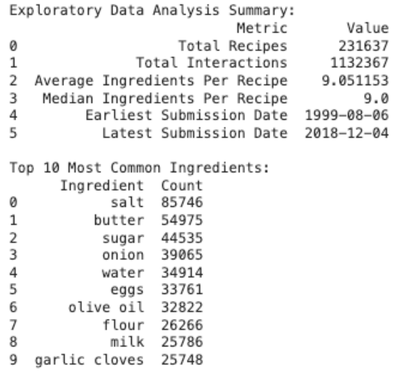
Dataset: Food.com Recipes and Interactions
Source:
https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions

This dataset includes over 230K+ recipes and 1.1M+ recipe reviews spanning 18 years of user interactions and uploads on Food.com. For our analysis, we will utilize two files: RAW_interactions.csv and RAW_recipes.csv. We selected this dataset due to its rich and diverse information. Some of the interesting values included:

- **`rating`**`: Rating given from user (0~5 scale)`
- **`review`**`: Review of the recipe (text)`
- **`minutes`**`: Minutes to prepare the recipe`
- **`nutrition`**`: Nutrition information (calories, total fat, sugar, sodium, protein, saturated fat, and carbohydrates)`
- **`n_steps`**`: Number of steps in a recipe`
- **`ingredients`**`: List of ingredient names`
- **`n_ingredients`**`: Number of ingredients`
- **`description`**`: User-provided description`

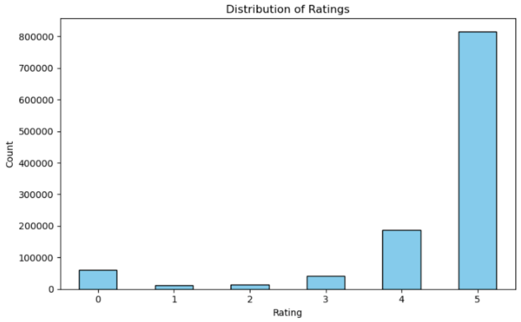
At first glance, a lot of values seemed to have interesting correlations to each other, such as 'minutes', 'n_steps', and 'n_ingredients' may affect the user's 'rating' because it could be a burden to cook a complicated recipe and may lead to a lower rating.

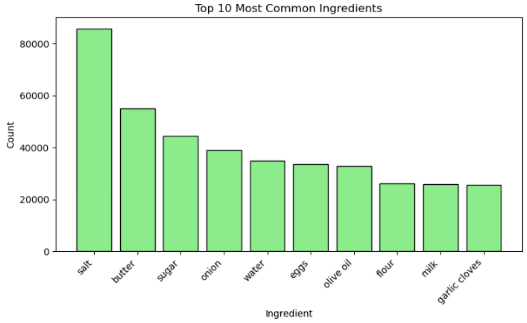
We have summarized the properties of the dataset and were able to retrieve the following results:

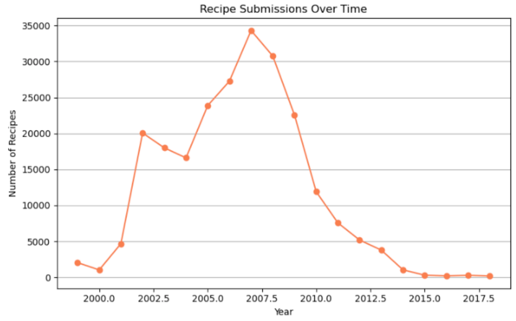


```
Exploratory Data Analysis Summary:
                          Metric       Value
0                  Total Recipes      231637
1             Total Interactions     1132367
2   Average Ingredients Per Recipe    9.051153
3    Median Ingredients Per Recipe         9.0
4         Earliest Submission Date  1999-08-06
5           Latest Submission Date  2018-12-04

Top 10 Most Common Ingredients:
      Ingredient  Count
0           salt  85746
1         butter  54975
2          sugar  44535
3          onion  39065
4          water  34914
5           eggs  33761
6      olive oil  32822
7          flour  26266
8           milk  25786
9   garlic cloves  25748
```







Analysis

Dataset Characteristics:

- The dataset is extensive with over 230,000 recipes and 1.1 million interactions, making it robust for machine learning tasks like recommendation systems or rating predictions.
- The average number of ingredients per recipe (around 9.55) suggests that most recipes are relatively simple, with a manageable number of ingredients.

Rating Distribution:

- The ratings are heavily skewed towards the highest score (5), which might indicate a bias in user reviews or a tendency for users to rate recipes they liked rather than disliked.
- The small number of ratings below 4 suggests that poorly received recipes may either not be reviewed often or that users prefer not to leave negative feedback.

Ingredients Popularity:

- Common ingredients like salt, butter, sugar, and onion dominate the dataset, reflecting their foundational role in a wide variety of recipes.
- These ingredients are versatile and likely to appear in both simple and complex recipes, which might impact model predictions for recipe similarity or ingredient importance.

Temporal Trends:

- Recipe submissions peaked around 2007, reflecting either a surge in platform popularity or increased engagement during that period.
- The sharp decline after 2007 might be due to changes in platform usage, competition from other recipe-sharing platforms, or a reduction in user engagement.

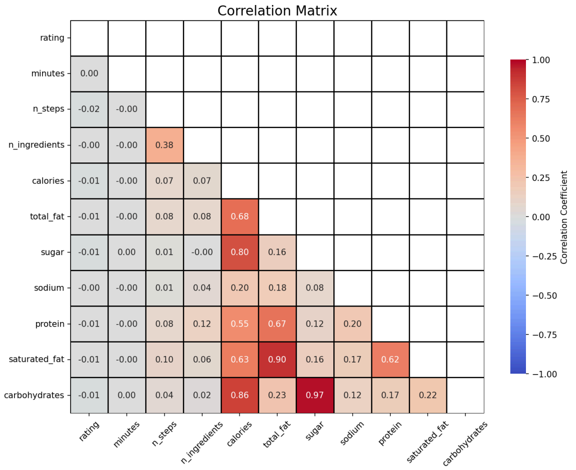
Insights for Predictive Modeling:

- The skewed rating distribution could pose challenges for regression models, requiring careful handling, such as rebalancing or weighted loss functions, to mitigate bias.
- The dominance of certain ingredients might mean they have less predictive power for ratings, as they are present in a large proportion of recipes.
- Temporal trends could be factored into models to study how recipe popularity or user engagement changed over time, which might also correlate with rating patterns.

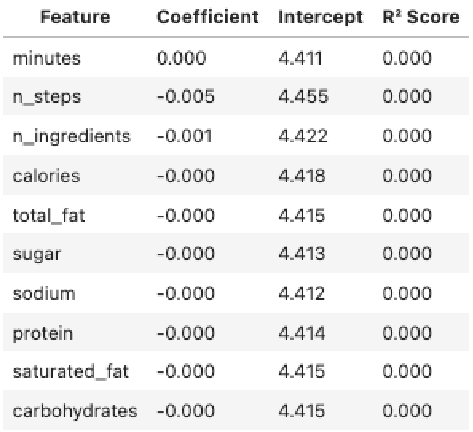
2. Predictive Task and Evaluation

Predictive Task 1

The goal is to predict the rating. As a first step, we analyzed whether any recipes' features correlate with the rating. We examined all the numerical features—`minutes`, `n_steps`, `n_ingredients`, and all the nutrition (calories, total fat, sugar, sodium, protein, saturated fat, and carbohydrates)—to determine if they have any correlation with the rating.



In addition to calculating the correlation, we also determined the coefficients to quantify the relationship between these features and the rating:

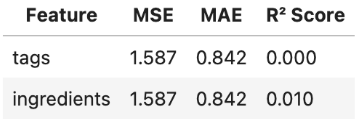


Feature	Coefficient	Intercept	R² Score
minutes	0.000	4.411	0.000
n_steps	-0.005	4.455	0.000
n_ingredients	-0.001	4.422	0.000
calories	-0.000	4.418	0.000
total_fat	-0.000	4.415	0.000
sugar	-0.000	4.413	0.000
sodium	-0.000	4.412	0.000
protein	-0.000	4.414	0.000
saturated_fat	-0.000	4.415	0.000
carbohydrates	-0.000	4.415	0.000

The analysis shows that minutes, n_steps, n_ingredients, and all the nutrition values have negligible correlations with rating, and their regression models exhibit extremely low coefficients and R^2 scores of 0.000, indicating no explanatory power for variance in rating.

Next, we turn our attention to tags and ingredients. Each recipe contains a list of tags and ingredients, which are challenging to evaluate directly. To simplify the analysis, we converted these tags and ingredients into vectors. Now, we will examine whether these vectors show any correlation with rating.

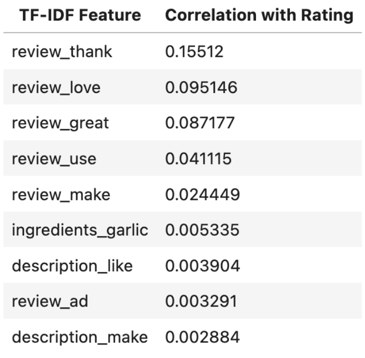
For tags and ingredients, which were converted into vectors, we ran a regression model and evaluated it using MSE, MAE, and R² scores to determine whether these vectors are correlated with rating. For ingredients, due to the large number of unique entries, we filtered the dataset to include only ingredients that appear more than 500 times (579 ingredients) before running the regression.



Feature	MSE	MAE	R² Score
tags	1.587	0.842	0.000
ingredients	1.587	0.842	0.010

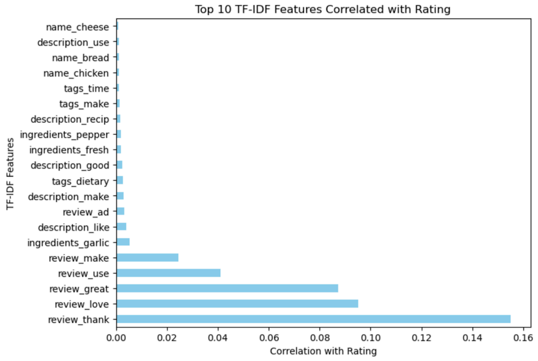
The regression analysis for tags and ingredients vectors shows negligible correlation with rating. Both features resulted in high MSE (1.587) and MAE (0.842), indicating poor prediction accuracy. The R² scores are near zero (tags: 0.000, ingredients: 0.010), confirming that these vectors do not significantly explain the variance in rating, even after filtering for frequently used ingredients.

Lastly, we analyzed text data, including review, tags (text), ingredients (text), name, and description. The text was first converted into TF-IDF vectors, and linear regression was applied using one-hot encoding for all the text data.



TF-IDF Feature	Correlation with Rating
review_thank	0.15512
review_love	0.095146
review_great	0.087177
review_use	0.041115
review_make	0.024449
ingredients_garlic	0.005335
description_like	0.003904
review_ad	0.003291
description_make	0.002884

To evaluate the relationship between the text features and rating, we computed the correlation matrix to identify the most significant textual features. Among all the features, review text showed the highest correlation with rating, with words like “thank,” “love,” and “great” having the strongest positive associations. Other features, such as tags, ingredients, name, and description, showed minimal or negligible correlations.



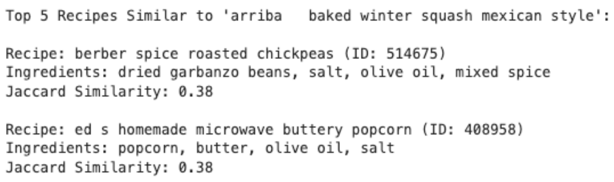
Since only the review text shows meaningful correlations with ratings, we decided to build the rating prediction model using only the review text.

Predictive Task 2

This task is to recommend recipes similar to a given recipe based on their ingredient similarity. The Jaccard Similarity is used to calculate the similarity between the sets of ingredients in recipes. The results provide the top 5 similar recipes for a given target recipe.

The ingredients of each recipe are used as input features. These are stored in the ingredients column in the dataset. The ingredients column is converted into a set for each recipe, enabling the computation of Jaccard Similarity. Recipe IDs are mapped to their respective ingredient sets using the id column for identification.

Jaccard Similarity Measures the similarity between two sets by comparing the ratio of the intersection size to the union size. The baseline for this task is a random recommendation, which serves as a benchmark to compare the performance of the Jaccard-based similarity recommendation system.



```
Top 5 Recipes Similar to 'arriba   baked winter squash mexican style':

Recipe: berber spice roasted chickpeas (ID: 514675)
Ingredients: dried garbanzo beans, salt, olive oil, mixed spice
Jaccard Similarity: 0.38

Recipe: ed s homemade microwave buttery popcorn (ID: 408958)
Ingredients: popcorn, butter, olive oil, salt
Jaccard Similarity: 0.38
```

Ingredient Overlap: Many of the recommended recipes have overlapping ingredients like salt, olive oil, and butter, which are common foundational ingredients. This high overlap in foundational ingredients likely drives the similarity scores. The recommendations are heavily influenced by the most commonly occurring ingredients (e.g., salt, olive oil), which could reduce diversity in recommendations. Recipes with unique or specialized ingredients may not be well-represented due to their rarity in the dataset.

3. Model Description

Predictive Task 1

We refined our objective to focus exclusively on the review text, as it proved to be the only relevant feature for predicting ratings. Our updated goal is to develop a robust rating prediction model based solely on review text.

To achieve this, we processed the review text into TF-IDF vectors and trained a linear regression model using one-hot encoding. To prevent overfitting, we applied regularization techniques, specifically Lasso and Ridge regression, and fine-tuned the models by performing a grid search to identify the optimal parameters for each regularization method. Additionally, we experimented with varying the number of top N words used in the TF-IDF vectorization to evaluate the impact on model performance. The training data comprised 1.12 million reviews, providing a comprehensive dataset for robust model training and evaluation.

Baseline model:

For comparison, we established a baseline model that predicts all ratings as the average rating of the training dataset.

Metrics:

To evaluate the performance of our models, we used fixed test data consisting of 10,000 reviews and measured metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R² score. These metrics allowed us to quantify how well our models performed compared to the baseline and assess the effect of different configurations, such as regularization parameters and the number of TF-IDF features used.

Predictive Task 2

Predicting user ratings directly as a numeric value proved challenging due to limited relationships between features and review text and the vast variety of text entries. Therefore, we reframed the task as a classification problem: predicting whether a user will give a high rating (4 or more) to a recipe.

We used a Random Forest Classifier for its ability to handle non-linear relationships, robustness against overfitting, and interpretability through feature importance analysis. The decision to use Random Forest was influenced by its effectiveness in handling mixed data types and its scalability to large datasets.

```
Features (X) = [
     "jaccard_similarity": Jaccard similarity
score between the ingredients of the given
recipe and the ingredients of all recipes
interacted by the user.
     "num_ingredients": Number of ingredients
in the recipe.
     "avg_rating": Average score for the
recipe across all users.
     "num_reviews": Number of reviews for the
recipe.
     "user_avg_rating": The user's average
score for all their reviews.
     "user_total_interactions": Total number
of recipes the user has reviewed.
]
Target (Y) = "high_rating": Binary
classification target, where: 1 = High rating
(4 or more), 0 = Low rating (less than 4)
```

Optimization and Implementation

- The Random Forest Classifier was trained with default hyperparameters but could be optimized further using grid search or random search for:
    - Number of trees (n_estimators)
    - Maximum tree depth (max_depth)
    - Minimum samples per leaf (min_samples_leaf)
- The model was evaluated using a train-test split (80% training, 20% testing).

Feature Importance:

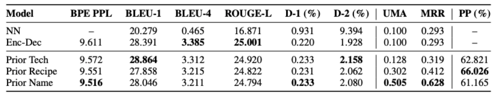
1. Top Features:
- user_avg_rating: Strongest predictor, showing the influence of a user's historical behavior.
- jaccard_similarity: Highlights the relevance of ingredient alignment between recipes and user preferences.
- num_reviews: Recipes with more reviews tend to be more predictable in terms of rating.
2. Other Features:

- avg_rating and user_total_interactions also contribute meaningfully, while num_ingredients was less significant.

4. Related Literature

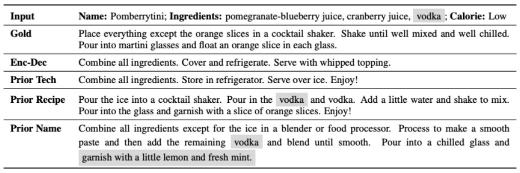
Dataset Origin and Usage: We have found two studies that utilized the Food.com dataset, which contains over 180K recipes and 700K user interactions, spanning 18 years. The dataset offers a wealth of textual and interaction data, including recipe names, ingredients, steps, and reviews.

Existing Work: The first study, 'Generating Personalized Recipes from Historical User Preferences' by Majumder et al. (supervised by Prof. Julian McAuley) proposed a personalized recipe generation model that leverages user history. They introduced a novel dataset split, where recipes are generated based on partial inputs and historical preferences. This method outperformed traditional encoder-decoder baselines in personalization and coherence. They have introduced attention-based mechanisms (e.g., Prior Recipe, Prior Name) to capture user preferences for recipe generation. Also, they achieved improved coherence and personalization metrics, with "Prior Name" attention outperforming other techniques.



Model	BPE PPL	BLEU-1	BLEU-4	ROUGE-L	D-1 (%)	D-2 (%)	UMA	MRR	PP (%)
NN	–	20.279	0.465	16.871	0.931	9.394	0.100	0.293	–
Enc-Dec	9.611	28.391	**3.385**	**25.001**	0.220	1.928	0.100	0.293	–
Prior Tech	9.572	**28.864**	3.312	24.920	0.233	**2.158**	0.128	0.319	62.821
Prior Recipe	9.551	27.858	3.215	24.822	0.231	2.062	0.302	0.412	**66.026**
Prior Name	**9.516**	28.046	3.211	24.794	**0.233**	2.080	**0.505**	**0.628**	61.165

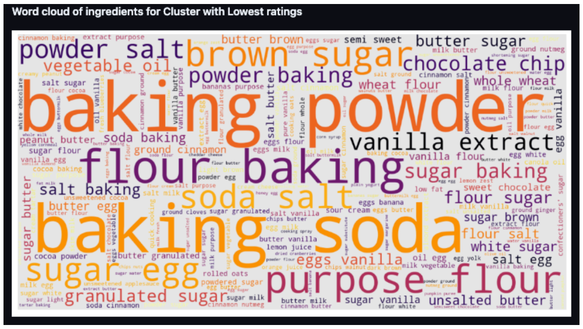
Table 2: Metrics on generated recipes from test set. D-1/2 = Distinct-1/2, UMA = User Matching Accuracy, MRR = Mean Reciprocal Rank, PP = Pairwise preference over baseline (evaluated for 310 recipe pairs per model).



Input	**Name:** Pomberrytini; **Ingredients:** pomegranate-blueberry juice, cranberry juice, vodka ; **Calorie:** Low
Gold	Place everything except the orange slices in a cocktail shaker. Shake until well mixed and well chilled. Pour into martini glasses and float an orange slice in each glass.
Enc-Dec	Combine all ingredients. Cover and refrigerate. Serve with whipped topping.
Prior Tech	Combine all ingredients. Store in refrigerator. Serve over ice. Enjoy!
Prior Recipe	Pour the ice into a cocktail shaker. Pour in the vodka and vodka. Add a little water and shake to mix. Pour into the glass and garnish with a slice of orange slices. Enjoy!
Prior Name	Combine all ingredients except for the ice in a blender or food processor. Process to make a smooth paste and then add the remaining vodka and blend until smooth. Pour into a chilled glass and garnish with a little lemon and fresh mint.

Table 3: Sample generated recipe. Emphasis on personalization and explicit ingredient mentions via highlights.

The second study, 'Text Analytics on Food.com Recipes/Review Data'" by Pavan Kommareddy explored text analytics on the same dataset, focusing on vectorizing review text with techniques like TF-IDF, Word2Vec, and sentiment analysis. It classified recipes into cuisines using external datasets and employed clustering for insights into ingredient-based patterns. This study highlighted user preferences for low-sugar, high-protein recipes through clustering. He implemented XGBoost and neural networks for sentiment classification, achieving an AUC of 78.5%. He also classified recipes into cuisines with an 80% accuracy using TF-IDF vectorization. He then created word clouds for ingredients in the most popular and least popular clusters to find any interesting patterns to visualize the results.



Comparison with Our Work: Our rating prediction model complements these studies by focusing on predicting ratings from reviews and user preferences using features like ingredient similarity and user interaction history. Unlike Majumder et al., our approach does not generate recipes but instead predicts user ratings, providing insights into user preferences through feature importance analysis. The clustering insights from the second study align with our findings on user rating behavior, emphasizing health-conscious choices.

In summary, while prior studies emphasized recipe generation and sentiment analysis, our work bridges the gap by directly predicting ratings and user preferences, thereby contributing to understanding user-recipe dynamics.

5. Results and Conclusions

Results

Predictive Task 1:

The baseline model, which predicts all ratings as the average rating, yielded the following performance metrics:

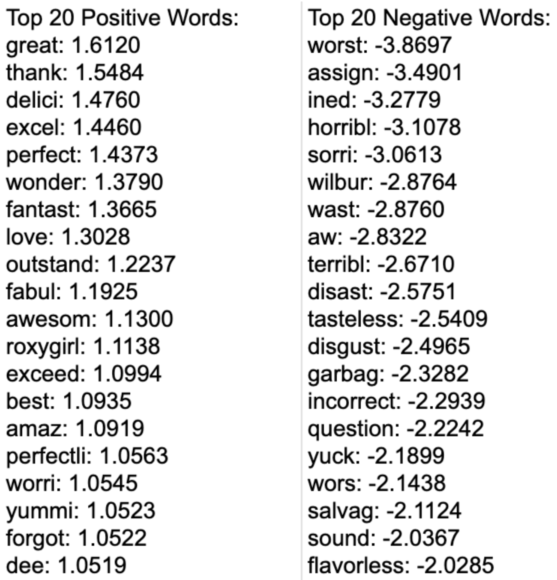
```
Mean Square Error (MSE): 1.562
Mean Absolute Error (MAE): 0.8421
R² score: -0.0000518
```

In comparison, Ridge regularization outperformed Lasso regularization, providing better results in terms of prediction accuracy.

The optimal model was achieved with Ridge regularization using an alpha value of 10.0 and top N words set to 8000. The performance metrics for this model were:

```
Mean Square Error (MSE): 1.2823
Mean Absolute Error (MAE): 0.7039
R² score: 0.1793
```

Further increasing the top N words beyond 8000 showed minimal improvement in metrics, indicating that expanding the vocabulary size beyond this point is unlikely to yield significant gains in performance. Most words had little influence on the rating, but some had a substantial impact. The coefficients of stemmed words that had the most positive and negative effects on the rating are as follows:



```
Top 20 Positive Words:        Top 20 Negative Words:
great: 1.6120                 worst: -3.8697
thank: 1.5484                 assign: -3.4901
delici: 1.4760                ined: -3.2779
excel: 1.4460                 horribl: -3.1078
perfect: 1.4373               sorri: -3.0613
wonder: 1.3790                wilbur: -2.8764
fantast: 1.3665               wast: -2.8760
love: 1.3028                  aw: -2.8322
outstand: 1.2237              terribl: -2.6710
fabul: 1.1925                 disast: -2.5751
awesom: 1.1300                tasteless: -2.5409
roxygirl: 1.1138              disgust: -2.4965
exceed: 1.0994                garbag: -2.3282
best: 1.0935                  incorrect: -2.2939
amaz: 1.0919                  question: -2.2242
perfectli: 1.0563             yuck: -2.1899
worri: 1.0545                 wors: -2.1438
yummi: 1.0523                 salvag: -2.1124
forgot: 1.0522                sound: -2.0367
dee: 1.0519                   flavorless: -2.0285
```

Predictive Task 2:

After training a random forest model with features including ingredient-based Jaccard similarity, we observed:

```
Evaluation Metrics:
Accuracy: 0.92
Precision: 0.93
Recall: 0.98
F1 Score: 0.96
```

These results indicate that the model performs well in predicting whether a user will rate a given recipe highly. The importance of each feature shows the contribution of each feature to the model:

```
Feature Importance:
user_avg_rating: 0.478854
jaccard_similarity: 0.142979
num_reviews: 0.122781
avg_rating: 0.119761
user_total_interactions: 0.082831
num_ingredients: 0.052793
```

- The most influential feature is the user's average rating across recipes. This suggests that a user’s general rating behavior (e.g., whether they are typically lenient or strict) significantly impacts the predictions.
- While not the top feature, Jaccard similarity contributes substantially (14.3% importance), which may indicate that personalized ingredient matching helps refine the recommendations.

Conclusion

The analysis revealed key insights into predicting recipe ratings. In Predictive Task 1, we aimed to predict ratings using numerical features such as minutes, n_steps, n_ingredients, and nutrition values (e.g., calories, sodium, sugar). These features exhibited negligible correlations with ratings, and their regression models demonstrated minimal coefficients and R² scores close to zero, indicating no explanatory power for predicting ratings. In contrast, review text emerged as the only feature significantly correlated with ratings. Using TF-IDF vectorization of review text combined with Ridge regression, the optimal model was achieved with an alpha value of 10.0 and top N words set to 8000, yielding a Mean Squared Error (MSE) of 1.2823, Mean Absolute Error (MAE) of 0.7039, and an R² score of 0.1793. These findings underscore the importance of textual data, particularly review text, for rating prediction, with diminishing returns observed when expanding the vocabulary size beyond 8000 words.

In Predictive Task 2, the focus shifted to predicting high recipe ratings (4 or more) using a Random Forest Classifier. The analysis demonstrates that user-specific features, particularly the user’s average rating (user_avg_rating), are significant predictors of recipe preferences, contributing nearly half of the model’s predictive power. Personalized ingredient alignment through Jaccard similarity added value, accounting for 14.3% of feature importance. This indicates that aligning recipes with a user's ingredient preferences enhances recommendation quality.

General recipe attributes, such as the number of reviews (num_reviews) and average rating (avg_rating), provided additional context but were less influential than user-specific features. The limited impact of recipe complexity, as measured by the number of ingredients (num_ingredients), suggests that users prioritize other factors over simplicity.

With an F1 score of 96%, the Random Forest model effectively balances user behavior, recipe characteristics, and personalization. Future improvements could include integrating additional user-specific features, such as cuisine preferences or dietary needs, as well as contextual factors like seasonal trends, to further enhance prediction accuracy and recommendation quality.

Together, these tasks demonstrate the critical role of textual data and user behavior in understanding and predicting recipe preferences, paving the way for more personalized and accurate recommendation systems.

References

Food.com Recipes and Interactions (Kaggle), "Crawled data from Food.com (GeniusKitchen) online recipe aggregator," [Online]. Available: https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions?select=PP_recipes.csv.

B. P. Majumder, S. Li, J. Ni, and J. McAuley, "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), Hong Kong, China, 2019, pp. 5975–5981. [Online]. Available: https://aclanthology.org/D19-1613.

P. Kommareddy, "Text Analytics on Food.com Recipes/Review Data," GitHub Repository. [Online]. Available: https://github.com/kbpavan/Text-Analytics-on-Food.com-Recipes-Review-Data-?tab=readme-ov-file.