Recommender Systems on Food.com Rating

1. Introduction of the background

In today's fast-paced life, cooking at home is not only an economical choice but also a crucial means to promote a healthy lifestyle and family gatherings. Through home cooking, we can better control the quality and nutritional value of ingredients, contributing to the development of balanced dietary habits and overall well-being.

In this process, the significance of recipes cannot be overstated, and Food.com stands out as an ideal platform providing practical guidance for those who wish to cook at home. Food.com boasts an extensive recipe database covering a variety of tastes and difficulty levels, catering to both beginners and experienced chefs alike.

This website not only offers a wealth of recipes but also provides detailed cooking instructions, ingredient lists, and other kitchen tips. Users can benefit from browsing through reviews and comments from other chefs, gaining practical cooking advice to enhance the smoothness of the cooking process.

For websites like Food.com, recommendation algorithms play a crucial role in understanding user preferences and improving recipe recommendations on the website. To further enhance the user experience, we have decided to employ three different models to predict ratings accurately. The first model relies on users' past usage history and recipe data to predict ratings. The second model uses users' recipe rating records and reviews as a basis for predicting the final recipe rating. The last model is an improved version of the first, incorporating additional analysis and processing based on users' historical ratings, scores, and recipe data, aiming to achieve even better rating predictions.

2. Dataset

2.1 Dataset content

The first dataset used is the dataset of Raw_recipes, the statistics of the dataset is shown below.

Column	Description	Non-null Entries	Object
name	The name of the recipe	231636	object
id	The id of the recipe	231637	int64
minutes	Minutes that the recipe take	231637	int64
contributor_id	The contribute id of the recipe	231637	int64
submitted	The submitted day of the recipe	231637	object
tags	Tags food.com label the recipe	231637	object
nutrition	nutrition of the recipe (calories, total fat, sugar, sodium, protein, saturated fat)	231637	object
n_steps	Number Of steps of the recipe	231637	int64
steps	Number of steps of the recipe	231637	object
description	Description of the recipe	226658	object
ingredients	Ingredients of the recipe	231637	object
n_ingredients	Number of ingredients of the recipe	231637	int64

The dataset contains over 230k of recipe information in food.com.
The second dataset used is the dataset of RAW_interactions, the statistics of the dataset is shown below.
The dataset contains over 1.1 million of user information in food.com.

We first merged two datasets into a new dataset using "Raw_interaction" as the reference, resulting in a dataset with over 1.1 million entries. Subsequently, we observed a significant number of ratings

Column	Description	Non-null Entries	Object
user_id	The ID of the user	1132367	int64
recipe_id	The ID of the review recipe	1132367	int64
date	The date review was submitted	1132367	object
rating	Rating that the user given to the review recipe	1132367	int64
review	The review text for the recipe that the user submitted	1132198	object

with a value of 0. Since the minimum possible rating is 1, we treated these values as missing and replaced them with np.nan. Furthermore, we decomposed the "nutrition" column into individual components such as calories, total fat, sugar, sodium, protein, saturated fat, carbohydrates, minutes, n-steps, and n-ingredients for a more in-depth analysis.

The statistics of the new columns used in the combined and clean dataset is shown below.

Column	Description	Non-null Entries	Object
calories	The calories of the recipe	652716	float64
total_fat	The total fat of the recipe	652716	float64
sugar	The sugar of the recipe	652716	float64
sodium	The sodium of the recipe	652716	float64
protein	The protein of the recipe	652716	float64
saturated_fat	The saturated fat of the recipe	652716	float64
carbohydrates	The carbohydrates of the recipe	652716	float64
Before 2013	Whether the recipe is submitted before 2013(include 2013)	652716	bool
submitted_year	The submitted year of the recipe	652716	int64
step_over_10	Whether the recipe take over 10 steps(include 10)	652716	bool
ing_over_10	Whether the recipe take over 10 ingredients(include 10)	652716	bool

Upon obtaining the complete dataset, we initially employed the interquartile range method to identify and drop outliers from numeric columns, this removes errors in the data and makes our model more efficient and faster. Finally, we removed all rows with missing values (either empty or np.nan) across the entire dataset, resulting in the ultimate dataset containing approximately 650,000 entries.

2.2 dataset analysis

After processing and cleaning the dataset, we conducted a series of analyses to explore and identify unique characteristics within the data.

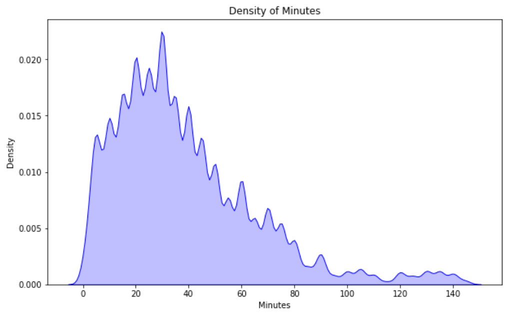

Figure 1: The Plot of Density of Cooking in Minutes

In Figure 1, we generated a chart to examine the distribution of the time required for each recipe. It is clear from the chart that the most common time for recipes falls within the range of 20 to 40 minutes and fewer recipes take longer, up to around 140 minutes.

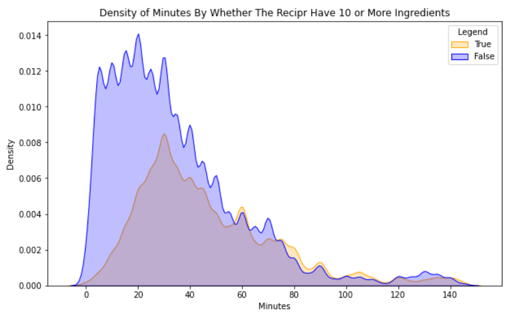

Figure 2: The Plot of Density of Minutes By Whether the Recipe Has 10 or More Ingredients.

We group recipes to two groups, the first group is recipes that take 10 or more ingredients, shown as orange on the graph. Else, those recipes take less than 10 ingredients, shown as blue on the graph. The key observation is that recipes with fewer than 10 ingredients typically take less time to prepare, in contrast, recipes with 10 or more ingredients have a wider spread of preparation times. This suggests that more complex recipes with more ingredients generally require more preparation time.

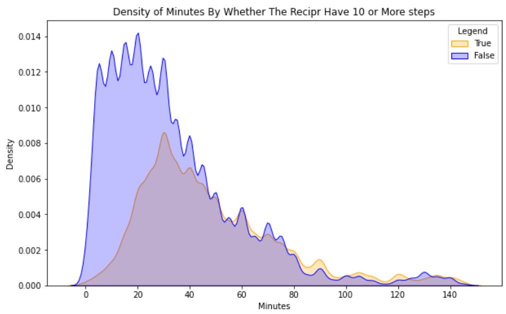

Figure 3: The Plot of Density of Minutes By Whether the Recipe Has 10 or More steps.

We group recipes to two groups, the first group is recipes that take 10 or more steps, shown as orange on the graph. Else, those recipes take less than 10 steps, shown as blue on the graph. The most important observation is that recipes with fewer than 10 steps (the blue curve) tend to have a higher concentration of shorter preparation times, in contrast, recipes with 10 or more steps (the orange curve) need more time to prepare. This suggests that more complex recipes with a greater number of steps may require more time to complete.

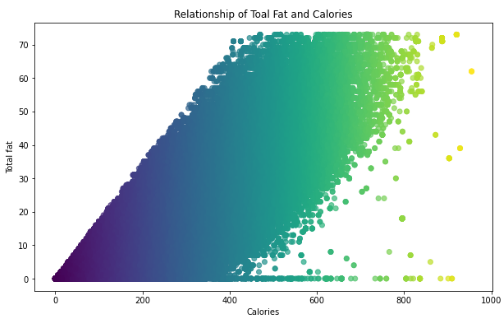

Figure 4: The Plot of Relationship of Total Fat and Calories

Figure 4 well interprets the relationship between total fat and calories, it shows that there is a very obvious positive correlation between total fat and calories, when total fat increases, calories will also increase together, which is in line with common sense. And we can also find that the density in the graph is higher in low fat and low calories, which in turn means that there are more low oil and low calorie recipes, possibly due to food.com's user preference.

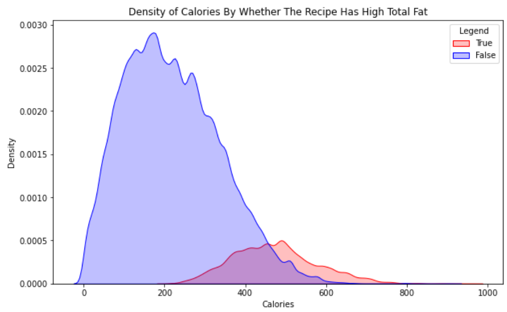

Figure 5: The Plot of Density of Minutes By Whether The Recipe Has High Total Fat.

To further analyze the fat and calories, we have divided the total fat into two groups, high total fat and low total fat. Based on Figure 5, the most important observation is that recipes that are high in total fat tend to have a higher calories and density distributed at the right of the graph. In contrast, recipes that are low in total fat tend to have lower calories and are distributed left of the plot. This gives a general idea of how calories are distributed in recipes in different total fat.

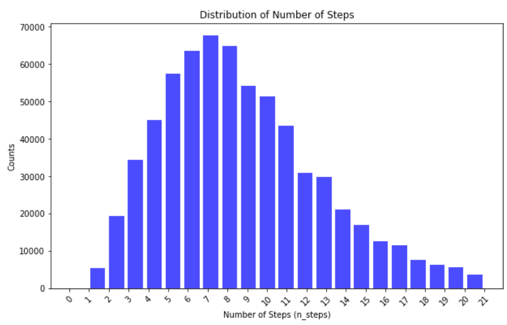

Figure 6: The Plot of Distribution of Number of steps

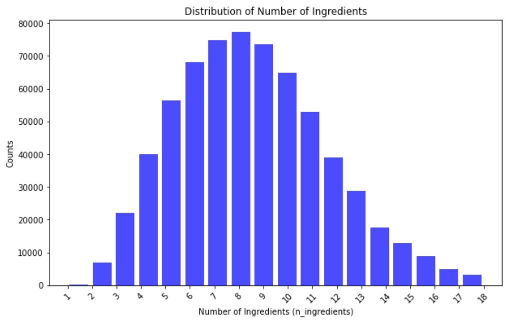

Figure 7: The Plot of Distribution of Number of Ingredients

The two histograms depict the distributions of the number of steps and the number of ingredients in recipes, respectively.

In the first histogram, which illustrates the number of steps, we observe a right-skewed distribution with the majority of recipes having between 6 to 10 steps. The counts decrease as the number of steps increases, indicating that recipes with a very high number of steps are less common.

The second histogram, showing the number of ingredients, also presents a right-skewed distribution with most recipes requiring between 7 to 10 ingredients. Similar to the first histogram, the frequency of recipes decreases as the number of ingredients increases, suggesting that recipes with a large number of ingredients are rarer.

Combining observations from both graphs, it can be inferred that most recipes are designed with a moderate number of steps and ingredients, optimizing for simplicity and convenience. Recipes that require many steps or ingredients are less common, potentially reflecting the preference for less complex cooking processes among the user for food.com.

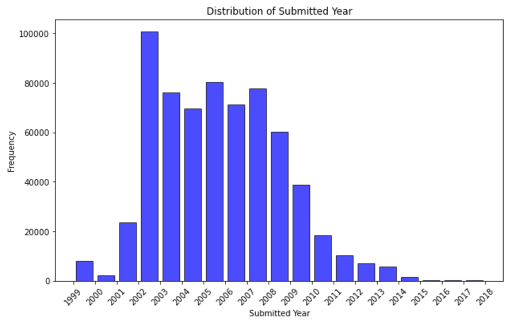

Figure 8: The Plot of Distribution of Submitted Year

In Figure 8 it can be observed that recipes have the most significant submitted quantities between the years 2002 and 2007, and peak in 2002, after which they begin to decline slowly. Until 2009, there was a break in the number of submissions and it started to decrease rapidly, and after 2016 it is even lower than in 2000. This may be due to the decline of food.com as the growth of the internet has led to more platforms for users and uploaders to choose from.

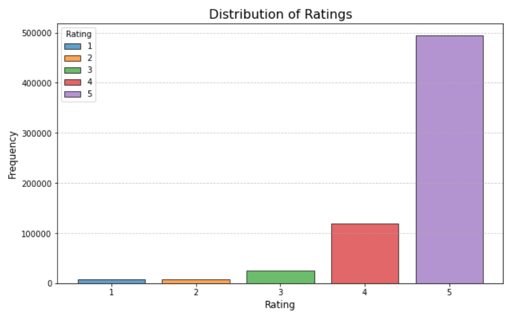

Figure 9: The Plot of Distribution of Rating

In order to make various predictions about ratings, we must first understand the distribution of each score in the rating, and Figure 9 explains this well. We can first easily find that 5-star ratings have the most significant distribution, far exceeding all the remaining ratings. Moreover, the frequency of ratings diminishes as the rating value decreases, and there is a positive correlation between the number of ratings and the number of ratings. This phenomenon may be caused by the fact that users do not have high requirements for recipes. As long as they can follow the rules or reproduce the taste correctly, they will not give low ratings even if they do not completely like the taste. Or users who don't like the recipe or have a bad experience tend not to rate or leave comments. As a result, the raters are users who are willing to give high scores and have a good experience. Therefore, 5 stars account for the vast majority, which suggests that this dataset may be biased.

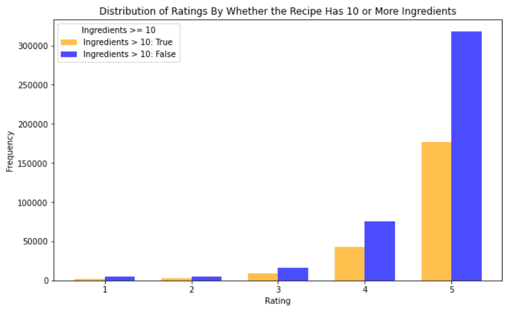

Figure 10: The Plot of Distribution of Rating By Whether the Recipe Has 10 or More Ingredients

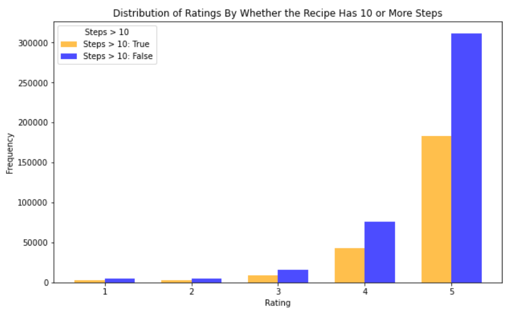

Figure 11: The Plot of Distribution of Rating By Whether the Recipe Has 10 or More Steps

The two histograms Figure 10 and Figure 11 represent the distribution of ratings for recipes, categorized by whether the recipes have 10 or more ingredients or steps.These two graphs have very similar distribution. It can be noticed that all the recipes with fewer ingredients and fewer steps have more frequency in both charts, from one star to five stars. This may mean that low steps and low ingredients recipes are more popular among food.com users, and therefore users who are willing to rate leave more ratings on these low process recipes.

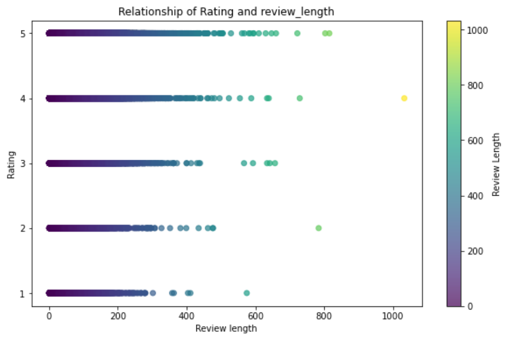

Figure 12: The Plot of Distribution of Rating By Review Length

Scatter plot Figure 12 shows the relationship between review length and rating, we can see that most of the reviews are distributed within 200 words, while the amount of higher length reviews decrease as the review length increases. Although the limited amount of data does not allow for definitive conclusions to be drawn regarding the positive correlation between reviews and ratings, this suggestive relationship warrants further validation and analysis.

3. Predictive Model

3.1 Introduction to the Task

Our task focuses on predicting user ratings for unseen recipes on the recipe-sharing platform Food.com. This endeavor aims to enhance the accuracy of recipe recommendations and personalized features. By utilizing users' past rating records, we can provide a more customized and engaging user experience, thereby boosting user engagement.

We primarily employed three different models for predicting ratings. Firstly, the Linear Regression User-Item Baseline Model captures users' rating preferences and the inherent characteristics of recipes. Secondly, the Sentimental Model considers users' emotional responses to recipes. The third model is SVD++, which can uncover more subtle interactions between users and recipes.

To keep these models performing well on unknown data and having good generalizability, we split the entire dataset into training, validation, and test sets in a 75%, 15%, and 15% ratio, respectively. Also, to ensure the accuracy of our predictions, we employed cross-validation. Our evaluation metric is the Mean Squared Error (MSE). In addition to comparing with the baseline model, we also compare our results with a naive model that always predicts the average rating. This comparison provides a direct understanding of whether the features we used effectively enhance the accuracy of our predictions.

3.2 Baseline Model

To predict users' ratings for unknown recipes, we use the Linear Regression User-Item Baseline Model as our baseline model with the formula rating(u,i) = $\alpha + \beta u + \beta i$. In this model, we use user_id and recipe_id from the data as features to capture the basic interactions between users and recipes. α represents the average of all ratings, providing a fundamental rating level for all predictions. βu is the user bias, indicating the difference between an individual user's ratings and the average rating, showing whether a user tends to give higher or lower ratings to recipes. Finally, βi is the item bias, reflecting the item's (recipe's) popularity relative to the average rating, indicating whether a recipe generally receives higher or lower ratings.

In this model, we initially set α as the average rating and initialize all user and item biases to zero. Following the iteration equation from our textbook, we use cross-validation on the validation set to find the optimal lambda. Through continuous iteration, we adjust α, βu, and βi until they converge, intending to minimize the difference between predicted and actual ratings. Finally, we assess the performance of the optimized model on the test set.

Employing cross-validation in our Baseline Model, we optimized the lambda value, which led us to achieve a mean squared error (MSE) of 0.476 on the test set. This result is slightly better than the naive model's MSE of 0.510, which always predicts the average rating. Although our baseline model can capture the overall trend, it is still a model that treats users and recipes independently and may not fully consider the interaction between users and recipes. This suggests that there is still much room for improvement in our model.

Model	Train MSE	Valid MSE	Test MSE
Baseline	0.3239206 58419596	0.4425396 85062272	0.4755313026 58443
naive (always average)	0.5095335 82908631	0.4822015 78994077	0.5102495927 1803

3.3 Sentimental Model

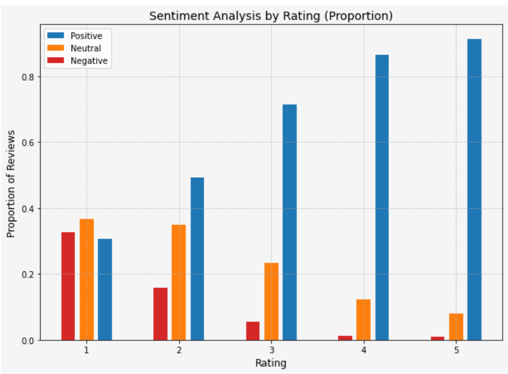

Figure 13: The Plot of Distribution of Rating By Proportion of Reviews Type

In our study, we extended beyond the limitations of our base model by exploring additional data columns that could enhance the functionality of our recommendation system. We discovered that the review columns, initially challenging to directly correlate with rating impacts, significantly influence customer rating behaviors. Our research confirms that both the emotional tone of review text and its perceived usefulness have a profound effect on readers' impressions and their likelihood of generating positive word-of-mouth. This impact is notably stronger among individuals with high product category involvement and those more susceptible to interpersonal influence, emphasizing the importance of detailed review analysis in predicting and understanding customer rating trends[1].

In our approach, we classified customer reviews into positive, neutral, and negative categories using TextBlob in python. Upon comparing lower and higher ratings, we observed an increase in neutral and negative reviews. This pattern led us to conclude that customer sentiments, as reflected in their written feedback, are intrinsically linked to their overall satisfaction levels. The prevalence of neutral and negative reviews at lower rating levels underscores the value of textual feedback in understanding customer rating behaviors.

In this model, we employed a sentiment analysis pipeline using Ridge regression to analyze customer reviews. We processed the textual data by normalizing the case and removing punctuation, followed by frequency analysis to identify key terms. We then constructed feature vectors for each review and segmented the dataset into training, validation, and test sets. We optimized the model by selecting an ideal lambda value through cross-validation, with the goal of minimizing the Mean Squared Error (MSE) on the validation set. We subsequently evaluated the performance of our optimized model on a test set.

During our optimal cross-validation process, we determined that the best lambda value was 100, where our model achieved the lowest validation MSE of 0.326. When tested, this optimized model yielded an MSE of 0.341, significantly lower than our base model's MSE. These findings indicate that incorporating sentiment analysis into our recommendation system substantially improves its accuracy in predicting user ratings, highlighting its advantage that it can glean insights from the textual content of user reviews, providing a deeper understanding of the users' emotional responses and subjective opinions about the items. However, its disadvantage is that it relies solely on textual data, which may not always accurately reflect the user's intended rating, and it can miss the broader behavioral patterns evident in their interactions with the items.

Model	Train MSE	Valid MSE	Test MSE
Sentimental	0.340089989 753126	0.326094030 911610	0.340749763 704155

3.4 SVD++

In advancing from our baseline model, we sought methods capable of capturing the nuanced interactions and preferences of our users. Our baseline model, while offering initial insights into user and recipe trends, did not account for the latent factors and subtle user behaviors that are integral to rating dynamics.

We thus turned to SVD++, an algorithm that enriches predictions with latent factors for both users and recipes, and notably, incorporates implicit interactions. Such interactions encompass how users engage with recipes beyond explicit ratings. Our model's equation utilizes a sophisticated matrix factorization approach, employing user and item latent vectors and integrating implicit feedback to predict ratings.

$\hat{r}(x,i) = \mu + Bx + Bi + qi^T(px + |Ix|^{-1/2} * \sum(j \in Ix)\ yj)$

In this equation, $\hat{r}(x,i)$ represents the predicted rating by user x for item i, with μ representing the global average rating. User and item biases are denoted by Bx and Bi, respectively. The terms qi, px, and yj refer to the item factor vector, the user factor vector, and the item factors for implicit feedback, respectively. Ix includes the set of items for which user x has provided implicit feedback, and |Ix| indicates the number of such items.

We employed Grid Search Cross-Validation to fine-tune SVD++'s hyperparameters, including latent factors, learning rates, and regularization terms. The model achieved a test MSE of 0.328, an improvement over our baseline. Nonetheless, the extensive feature consideration of SVD++ necessitates greater computational resources and elongated model construction time[2]. To address this, we experimented with dimensionality reduction techniques in interquartile range methods to improve training times. Despite these demands, SVD++ proves to be a highly effective model, surpassing our baseline in predictive capability.

Model	Train MSE	Valid MSE	Test MSE
SVD++	0.320643232803181	0.315125517490890	0.328305717212914

3.5 Final Model

To leverage the strengths of both models, we apply a linear regression on the combined outputs of sentiment analysis and SVD++ for rating predictions. This ensemble approach allows us to integrate the rich, subjective insights from sentiment analysis with the behavioral and pattern recognition strengths of SVD++, leading to a more accurate and holistic prediction model.

The mathematical representation of our ensemble approach is:
$\hat{r} = \alpha + \beta 1 * \text{SentimentScore} + \beta 2 * \text{SVDppScore}$

The coefficients α, $\beta 1$, and $\beta 2$ are the regression parameters where α is the intercept, and $\beta 1$ and $\beta 2$ are the weights assigned to the SentimentScore and SVDppScore, respectively. These weights are optimized during the training of the linear regression model to best fit the combined predictive power of sentiment analysis and SVD++ outputs to the actual ratings.

To optimize this ensemble model, we use a two-step approach. First, we tune the parameters of the sentiment analysis model and the SVD++ model separately. Once both models are individually optimized, we train a linear regression model to find the optimal combination of their outputs.

The advanced combined model achieved a test set MSE of 0.234, indicating a significant improvement and the highest accuracy among all models tested. In addition, this is close to the MSE of training and valid dataset, which shows that our model is not overfitting. The strengths of our ensemble model lie in its comprehensive approach and improved accuracy. Its weakness is the increased computational demand. Simpler models are more interpretable and less computationally intensive but may not capture all the nuances in the data.

Model	Train MSE	Valid MSE	Test MSE
Ensembled Linear Regression	0.235656495464549	0.230047725668990	0.234291902041452

4. Literature

The data we use is the Recipes and Interactions data from Food.com (GeniusKitchen) online recipe aggregator, a free recipe website, which is organized and put into Kaggle before being downloaded by us, contains all kinds of recipe information in food.com, ratings and reviews given by users, etc.

This data was originally used in the paper "Generating Personalized Recipes from Historical User Preferences"[3] to use users' historical data to help users create new recipes that are reasonable and personalized. The authors Majumder and Li used three models to approximate the goal in their study, namely, name-based Nearest-Neighbor model, Encoder-Decoder baseline with ingredient attention, and 3-personalized models, which combine the applications of Natural Language Processing and recommendation system to generate personalized recipe predictions for the goals.

The Natural Language Processing mentioned above allows machines to understand and interpret human language, and the analysis method that quantifies language and emotions is one of the best methods for analyzing this type of data.

Similar datasets like Amazon Question and Answer Data have been used in research[4] to build Moqa model, which leverages user review text to answer subjective and personalized queries. The conclusion drawn from their research, "Reviews proved particularly effective as a source of data for answering product related queries, this demonstrates the value of personal experiences in addressing users' queries." is very similar to the conclusion we obtained. User reviews, or the summarization of experiences, are of significant value in the prediction of data such as user comments and responses.

5. Conclusion

We explored several models focusing on different aspects of the user-item relationship and the textual data. These include the baseline Linear Regression User-Item model, a Sentiment Analysis model, the SVD++ model, and an ensemble approach combining sentiment analysis with SVD++ results through a linear regression model.

In our examination, we found that the user’s commented text feature has a significant impact on the rating prediction. In the sentimental model, we are able to achieve significant MSE improvement from the baseline. We are considering word frequencies as the features, and this improvement in result aligns with our exploratory analysis about the attitudes of the reviews texts. This further validates the relationship between the users emotions in the review text and their ratings.

In contrast, the baseline model that relied solely on basic user-item interactions failed to account for the latent factors and the rich context provided by user reviews. This omission resulted in less accurate and less personalized recommendations. However, it is compensated for by the use of SVD++. The latent features representing the underlying user and item characteristics extracted by the SVD++ model worked remarkably well. They captured the implicit preferences and behaviors of users, which aren't directly observable but have a significant impact on their rating patterns.

At the end, our results demonstrated that the ensemble model significantly outperformed the baseline and individual sentiment and SVD++ models. Specifically, the integration of sentiment analysis and SVD++ allowed for a more comprehensive understanding of user preferences, blending the qualitative data of user reviews text with the quantitative data of user-item interaction patterns. This approach enhanced prediction accuracy and provided a more personalized recommendation model.

References

[1] Patrick De Pelsmacker, Nathalie Dens and Alona Kolomiiets. 2017. The impact of text valence, star rating and rated usefulness in online reviews. https://doi.org/10.1080/02650487.2018.1424792

[2] Bam Bahadur Sinha and R. Dhanalakshmi. 2022. A systematic modelling of recommender system using derivative imitation of SVD and regression. https://doi.org/10.1109/IATMSI56455.2022.10119255

[3] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating Personalized Recipes from Historical User Preferences. https://doi.org/10.48550/ARXIV.1909.00105

[4] Alex Yang and Julian McAuley. 2015. Addressing Complex and Subjective Product-Related Queries with Customer Reviews. https://doi.org/10.48550/ARXIV.1512.06863