Recommender Systems Part

Hongying Tao

May 4, 2024

1 Introduction

This report explores the development and evaluation of a restaurant recommendation system in Evanston using extensive user reviews and restaurant data. Our analysis focuses on applying collaborative filtering and predictive modeling techniques to improve the accuracy of restaurant recommendations, aiming to reveal important patterns in consumer preferences and restaurant characteristics.

2 Data Preparation

The data for this project comes from two sheets in Excel: a `Reviews` sheet that contains restaurant reviews and a `Restaurants` sheet that records restaurant details.

First, to ensure data consistency, the `Restaurant Name` column in `Reviews` and `Restaurants` was standardized to correct inconsistent or incorrect spelling of restaurant names. For example, "*Claire's Korner*" is misspelled as "*Clare's Korner*" in the `Reviews`.

Next, by converting the `Restaurant Name` in the two sheet into sets and comparing them, we identified `Restaurant Name` that only appeared in the `Reviews` data. Among them, *World Market*, *Todoroki Sushi* and *La Principal* restaurants only appear in the `Reviews` data set, but there are no corresponding records in the `Restaurants` data set, so I removed them from the `Reviews` data. In addition, the data in the `Marital Status` column has been formatted and standardized, including removing spaces, adjusting case, and unifying terminology.

Finally, the `Reviews` and `Restaurants` are merged into a 1459 rows $\times$ 20 columns data frame through the `Restaurant Name` field to facilitate subsequent analysis and modeling.

3 Collaborative Filtering

3.1 Recommendation Base on Reviewer Feature

a) Unique reviewer feature matrix creation:

First, I removed columns from the dataset that were not directly related to demographics, keeping columns only `Reviewer Name`, `Birth Year`, `Marital Status`, `Has Children?`, `Vegetarian?`, `Weight (lb)`, `Height (in)`, `Preferred Mode of Transport`, `Northwestern Student?`. To ensure that each record in the data set is unique, duplicate user records

are removed through the `Restaurant Name` column.

Then I did a simple cleanup of the data in these columns. For numeric columns, such as `Birth Year`, `Weight (lb)`, and `Height (in)`, missing data are filled using the average value of the respective column; for categorical columns, such as `Marital Status`, `Has Children?`, `Vegetarian?`, `Preferred Mode of Transport`, and `Northwestern Student?`, I use the default `nan` to fill in the missing data, after that use the one-hot encoding method to convert categorical columns into numerical values for easy use in the model.

Finally, the processed numerical features and encoded categorical features are combined into a complete feature matrix, with each row representing the complete feature set of a unique user. The total number of unique user vectors is 1066 and the dimension of each vector is 19.

b) Reviewer feature similarity matrix:

Using the feature matrix created in part **a)**, calculate the cosine similarity between each user. This similarity matrix is converted into a DataFrame with the index and columns being `Reviewer Name`, which makes it possible to quickly look up similarity scores by `Reviewer Name`.

c) Recommendation Algorithm Performance

In the recommendation function, first find the similarity between this reviewer and other reviewer from the similarity matrix in part **b)** based on the provided reviewer name (for example, *Timothy Mace*). These similarities are sorted in descending order to find the most similar reviewer. For each similar reviewer, the algorithm finds its highest-rated restaurant. These restaurants are then returned as recommended results.

The Table 1 below shows the 3 restaurants that *Timothy Mace* recommended to him based on similar reviewer.

Table 1: Top 3 Restaurant Recommendations

Similar User	**Favorite Restaurant**
Enid Egan	Burger King
Anthony Grieco	Union Pizzeria
NU Student 12	Hokkaido Ramen

3.2 Recommendation Base on Restaurants Reviews

a) Reviewer and rating matrix creation::

To build the recommendation system, first I create a pivot table with `Reviewer Name` as the index, `Restaurant Name` as the column, and the value `Rating`. This tabular format is convenient for representing each user's ratings of different restaurants, bu t since not all users have visited all restaurants, many cells have missing values (`NaN`).

To fill in these missing values, I used a similarity matrix based on user feature from 3.1 to find the 5 users with the highest similarity to the currently rated user. For each missing

rating, I check the ratings of the same restaurant from these 5 similar users and calculate the average of these existing ratings as the filler value.

In practice, if some or all of these 5 similar users also lack ratings for a particular restaurant, these missing ratings are temporarily assumed to be 3. The choice of 3 as a default value is based on the midpoint of the rating scale (1 to 5), assuming a neutral stance toward the restaurant, indicating that the users neither particularly like nor dislike it. This neutral assumption helps avoid introducing bias into the dataset while ensuring the effectiveness and fairness of the recommendation system.

Through this method, I can effectively use existing user preference information to estimate missing ratings, thereby enhancing the recommendation system's ability to predict restaurants that users may like.

b) Recommendation Algorithm Performance:

In the recommendation function, I first use the user feature vectors created in part **a)**, calculate the cosine similarity between each user and all other users, thereby generating a new user similarity matrix. Next, for a specific user (e.g. *Sarah Belle*), I find the user in the matrix that is most similar to her. This time the algorithm found that the user most similar to *Sarah Belle* is *Juan Rogers*. Checking out *Juan Rogers*' highest rated restaurant, we found it was *Kuni's Japanese Restaurant* with a rating of 5.0 out of 5.0. Based on this discovery, the recommendation system recommended that Sarah Belle try this restaurant.

4 Predictive Modeling

4.1 Linear Regression Model

Data preparation

Feature Variables:

For numerical features, such as `Birth Year`, `Weight (lb)`, `Height (cm)`. These features were processed to handle missing values (filled with the mean) and standardized (Z-score normalization) to mitigate the impact of varying scales across different features.

For categorical features, `Marital Status`, `Has Children?`, `Vegetarian?`, `Preferred Mode of Transport`, `Northwestern Student?`, and `Cuisine`. These categorical features were transformed into numerical format through one-hot encoding, allowing them to be properly processed by the model.

Target Variable:

The Target variable is the users' ratings for restaurants, which the model aims to predict. It directly comes from the `Rating` column in the dataset, representing users' evaluative scores for various restaurants. This is a categorical numerical variable, specifically ranging from 1 to 5, reflecting the ordinal scale of ratings from poor to excellent.

Dataset split:

After the features and target variables are prepared, the entire data set is divided into a training set and a test set. I split the data set in an $80 - 20$ ratio, with 80% of the

data used to train the model and 20% of the data used to test the model's effect. And also set the random seed to 42 to ensure the consistency of data splitting and ensure the repeatability of the experiment.

Result

Model performance:

The linear regression model had an R^2 value of 0.0427 and a mean square error (MSE) of 1.416, showing very low prediction accuracy and reliability. These metrics indicate that the model explains only 4.27% of the variation in ratings, which is significantly low.

Stability:

Without fixing the random seed, the model performance showed extremely high instability, the R^2 value was even negative, and the prediction results were significantly different from the actual rating, almost completely shattering the accuracy of the model.

Name	Actual Rating	Predicted Score
Tony Pantoja	5	4.21875
Anna Pichler	5	3.625
Mike Albert	1	3.734375
Robert Ruka	1	4.890625
Debra Utter	5	2.828125

Table 2: Comparison of Actual and Predicted Ratings

4.2 Lasso regularization

Adjust the linear model from 4.1 to include Lasso regularization with an alpha value of 0.01.

Result

Model performance:

Although the R^2 value of the Lasso model is approximately 0.06147, which shows an improvement over the previous model's near-zero or negative R^2 values, the improvement is not substantial. However, the Mean Squared Error of the Lasso model is 1.966, which is higher compared to the standard model. This outcome is intriguing and warrants further investigation. Lasso regularization reduces model complexity by penalizing the absolute values of the coefficients, which typically increases the model's bias. While this can enhance the model's generalization ability on unseen data, it can also lead to an increase in errors, as evidenced by the higher MSE in our Lasso model.

Feature selection:

The Lasso model selected only 19 features out of a total of 43 features, and the coefficients of the other 24 features were reduced to zero, indicating that these features are not necessary for predicting restaurant ratings.

Features like `Weight (lb)` and `Height (cm)` have relatively small but non-zero coefficients, indicating that they have a slight impact on the score. Demographic characteristics such as `Marital Status_Widowed` and `Has Children?_No` have negative coefficients, indicating that these categories negatively affect score predictions. Like `Cuisine_Thai`

and `Cuisine_American` have strong negative coefficients, which may indicate that the interviewed users have a low preference for American and Thai food.

Table 3: The Coefficient of the Selected Feature

Feature	**Coefficient**
Cuisine_Burgers	-1.09428
Marital Status_nan	-0.849577
Vegetarian?_No	-0.518954
Cuisine_Thai	-0.424039
Cuisine_American	-0.345288
Cuisine_Coffee	0.278833
Cuisine_Italian	-0.265019
Cuisine_Mediterranean	-0.212184
Has Children?_No	0.201228
Marital Status_Widowed	-0.176639
Marital Status_Married	0.157478
Marital Status_Single	0.133472
Cuisine_South Asian	-0.114820
Cuisine_Chocolate	0.078841
Height (cm)	-0.067986
Cuisine_Bubble Tea	0.059871
Preferred Mode of Transport_Car Owner	0.057567
Vegetarian?_Yes	0.040525
Weight (lb)	0.001667

4.3 Linear Regression Model for Coffee Shop

Data preparation

The same as the linear model constructed in the previous section 4.1, this time only the data labeled as `"Coffee"` cuisine was analyzed. The data set is divided into a training set and a test set, with the test set accounting for 20% of the overall data.

Result

Model performance:

R^2 of the model is 0.16007, showing a relatively low but improved fit than the previous model, indicating the limited effectiveness of the model in explaining the variation in the data. The mean square error is 1.4412, a value that indicates a certain degree of error between the predicted value and the actual value.

Coefficient analysis:

Table 4 shows that `Has Children?_Yes` has the largest negative impact (coefficient of -0.74355), indicating that users with children may rate coffee shops lower. In contrast, the coefficient of `Preferred Mode of Transport_nan` is 0.493754, which is the largest positive influence feature, indicating that users who do not specify transportation preferences may have higher evaluations of coffee shops. Other positive features include `Has Children?_No` and `Vegetarian?_Yes` meaning child-free users and vegetarians generally

give higher ratings. `Marital Status_Single` and `Weight (lb)` also show positive coefficients, suggesting that users with single status and higher weight may be more satisfied with coffee shops.

Table 4: Feature Coefficients

Feature	**Coefficient**
Has Children?_Yes	-0.74355
Preferred Mode of Transport_nan	0.493754
Has Children?_No	0.435354
Vegetarian?_nan	-0.361062
Marital Status_nan	0.308196
Has Children?_nan	0.308196
Northwestern Student?_No	-0.300820
Northwestern Student?_Yes	0.300820
Preferred Mode of Transport_On Foot	-0.288127
Vegetarian?_Yes	0.271343
Marital Status_Single	-0.218292
Weight (lb)	0.171519
Preferred Mode of Transport_Public Transit	-0.156568
Birth Year	-0.134120
Height (cm)	0.102063
Marital Status_Married	-0.089904
Vegetarian?_No	0.089719
Preferred Mode of Transport_Car Owner	-0.049058

5 Text Embedding

Data preparation

To ensure data integrity and enhance the effectiveness of model training, I filled in the missing review text. Use a predefined review template that provides text descriptions based on different rating levels, from very negative (1 point) to very positive (5 points). This method is mainly used to fill in data items that have ratings but missing review text. The embedded text and rating data are still divided into a training set and a test set (the test set accounts for 20

Result

I found that the linear regression model using text embeddings performed best, with an MSE of 1.0001, significantly better than the traditional linear regression model of 1.4161 and the Lasso regression model of 1.9662. This shows that high-quality text embeddings significantly improve prediction accuracy, while Lasso regression may not fully utilize all the information in the text data due to the regularization process.

6 Something for Interesting

In my in-depth research into Evanston restaurant reviews, I stumbled upon the bizarre behavior of several unique food critics. First up was Jillan Dames, who reviewed 36 restaurants in one day, 35 of which she gave perfect scores – a staggering torrent of positive reviews! However, on

that same glorious day, Dennis Folse went in the exact opposite direction, giving 12 restaurants icy negative reviews.

Even more dramatic, we also found a reviewer named Castor Z who took his dissatisfaction with the Evanston Chicken Shack to new levels. In one day, Castor gave 15 extremely low 1-point reviews and angrily commented that if there was a zero-point option, he would choose it without hesitation. It seems that the feud between this restaurant and Castor is quite serious!

Finally, I also noticed Dennis Folse. His restaurant adventures were decidedly unhappy—he left a 1-point negative review on each of four restaurants. There was the same reason behind all the bad reviews - he thought the food in every restaurant was too spicy! Does this mean that Dennis has an unusually low tolerance for spiciness, or that these restaurants really go overboard with the seasoning?