Food.com Rating Prediction

University of California, San Diego
9500 Gilman Dr, La Jolla, CA 92093
CSE 158

ABSTRACT

This paper analyzes a set of data from the reviews of Food.com and uses the review text to predict the rating of a recipe from a user. For the prediction, it attempts to build a linear regression model and uses sentiment analysis with the Bag-of-words feature to predict the rating of each review.

INTRODUCTION

With the development of technology, online learning, work, and play have been integrated into our lives. We can find and learn different kinds of knowledge on the Internet. Eating food is an essential activity in our lives, so finding delicious recipes on the Internet to make food has become a very interesting online activity. A digital brand and online social networking service, Food.com offers recipes, food news, new and classic programming, and pop culture from home and celebrity chefs. Food.com can provide recipes, photos, articles, and video content on the Internet for other users to view, learn from, comment on, and exchange. Food.com has an extensive collection of recipes and many user reviews for different recipes. This paper collects and analyzes the possible relationship between existing user reviews text and ratings to build a predictor to predict ratings based on user review text.

1 DATASET

1.1 Identify a dataset to study.

Our dataset comes from Kaggle. The data set is RAW_interactions.csv in Food.com Recipes and Interactions. It contains 1,132,367 user reviews. We chose this dataset mainly because it has clear and easy-to-analyze labels and data. Each review data has the same user id, recipe id, date, rating, and review.
When using this data set, we use 1,100,000 review data as the training set to train the model, use 15,000 review data as the verification set to evaluate the performance of the model and select the best model, and use 15,000 review data as the test set to evaluate the performance of the model.r

1.2 Exploratory analysis

Feature name	***Feature Detail***
user_id	The number of user ID
recipe_id	The number of recipe ID
date	The date of interaction (YYYY-MM-DD)
rating	The number of rating that the user give to the recipe
review	The text of review that the user give to the recipe

Figure 1 The Frequency Could for user_id

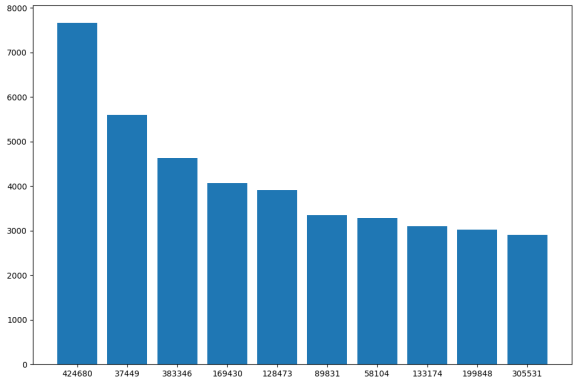

Figure 2 Bar for Top 10 Active User Reviews Count

First, we explore the user_id dataset. Figure 1 shows that not all users will review and have the same number of reviews using the cloud formed by the frequency of user_id. Therefore, having 1,132,367 review data doesn't mean there are 1,132,367 users. Through the exploration of the data set, we found that the data set has a total of 226,570 users. Active users will have reviews on different recipes, while inactive users may only appear in a small number of reviews. Figure 2 shows the distribution of user comments with the top 10 reviews. By exploring the dataset and the icons in Figure 2, we know that the maximum number of individual user reviews is 7,671, and the minimum number of user comments is 1. This shows that the data set can see that the distribution of user activity is not uniform. It well reflects the randomness of reviews collected by this data set. Therefore, this dataset is feasible as a dataset for research use.

Figure 3 The Frequency Could for recipe_id

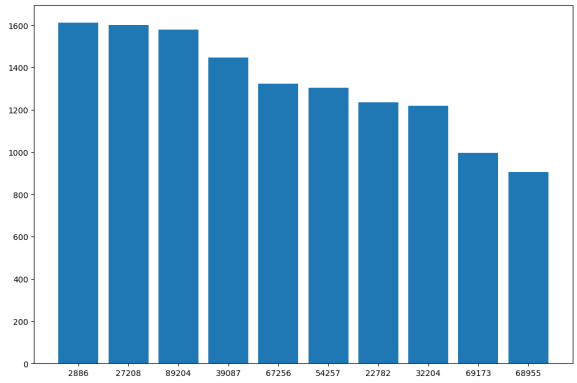

Figure 4 Bar for Top 10 Recipe Reviews Count

Next, we explore the recipe_id dataset. Similar to user_id, Figure 3 shows that not all recipes will have the same number of reviews by using the frequency cloud of recipe_id. So having 1,132,367 review data doesn't mean there are 1,132,367 recipes. By exploring the dataset, we found that there are a total of 231,637 recipes in the dataset that have reviews. A recipe may have a number of comments it has based on its content. Figure 4 shows the distribution of reviews for the top 10 recipes by the number

of reviews. Through exploration, we know that the maximum number of reviews a single recipe has is 1,613, and the minimum number of reviews a recipe has is 1. This shows that the data set can see that the distribution of recipes is not uniform. It reflects the randomness of the reviews collected by this data set. Therefore, this data set is feasible as a research data set.

Figure 5 The Frequency Could for date

Explore the dates in the dataset. According to the cloud formed by date appearance reviews (Figure 5), it can be seen that the distribution of dates is not even and has an extensive range. Through the exploration of the data, we know that the data set date is from 2000-01-25 to 2018-12-20; that is, the date range exceeds 18 years. Exploration and prediction of a large number of data in datasets are clearer and more efficient.

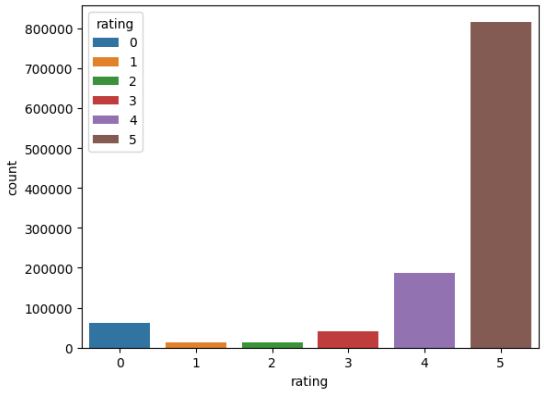

Figure 6 The count bar for rating

Next, we explore the distribution of ratings in the dataset. By counting the ratings and drawing a graph (Figure 6), it is interesting that there are far more reviews with a rating of 5 than the other ratings. A 4-point rating and a 1-point rating follow this. Ratings in the above graph are not evenly distributed, so what could cause the ratings to vary? Can we predict ratings from what features?

Figure 7 The Frequency Could for word in review

Finally, we explored the review data in the dataset. We try to look at the review and try to find the most frequently used words in the reviews. First, we try to remove the bad word from the review and then look for some high-frequency words in the review through the remaining vocabulary. Figure 7 is the word cloud we generated by removing

bad word reviews. From this figure, we can find that there are indeed some high-frequency words in the review. So does the use of high-frequency words affect the rating? We tried to study and build a predictor to see if there is a pattern.

2 PREDICTIVE TASK

Our prediction task is to predict the rating that the user will give based on the user's reviews. The predictor we built is of the form:

$$f(\text{text}) \rightarrow \text{rating}$$

using a model based on linear regression:

$$\text{rating} \simeq \alpha + \sum_{w \in \text{text}} \text{count}(w) \cdot \theta_w$$

We will use a part of the data as training data and read the reviews in the training data. There is no missing data in our data, so cleaning is unnecessary. There are a total of 1132367 data in the data set. We use the first 1,100,000 as the training set, 15,000 of the remaining data as the test set, and the other 15,000 as the test set.

For the length of the dictionary in the feature, we chose 2000. Note that we can use dictionaries of other sizes as well. But the larger the dictionary, the greater the runtime required. After using the training data to obtain the model, we will use the validation set to verify the accuracy of our model and analyze the MSE to determine whether the model is accurate enough. In addition, we use 0.3 as a threshold, and the difference between the predicted result and the real result by 0.3 is acceptable. An exact crude value can be obtained by counting the number of predictions within the threshold and dividing that by the total. If most of the data is predictive of success, then we can consider reviews to be a feature of rating.

Finally, we'll use our model to predict the test dataset. Now, the model can roughly predict a user's rating from a user's reviews.

3 MODEL

We use Regularized regression(linear_model.Ridge) as our machine learning model. We learn this model in week 6/7. Therefore, the mode we chose is the most suitable for our assignment 2. And we can also learn usage in Chapter 8 Workbook. Additionally, we will build a pipeline to test the optimal value of l. The range of the test is taken from this array: [0.01, 0.1, 1, 10, 100, 1000, 10000].

Currently, we use words in reviews as features. There is also time data in the data; we did not use this as a feature. Perhaps using time as a feature can improve the accuracy of the model.

3.1 Bag-of-words

Same as the Category prediction task in our assignment 1. We read all the reviews in the training set and split them using a Bag-of-words model. For bag-of-words N-grams, we just use simple unigrams. After sorting the bag of words, we can know the most frequently occurring words. We did not use stopwords and stemmer due to the long

running time. Accuracy and MSE should improve if we use stopwords and stemmers.

3.2 Sentiment analysis

After completing the construction of the bag of words, we can use the words in the bag of words for sentiment analysis. Similar to the method in assignment 1, we define a feature function which can compare the words in this review with the words in the bag-of-words when reading a new review. If the word is in the bag of words, then set this word as a feature.

-0.7863595009236012, 'sounds'
-0.7631447094763337, 'sorry'
-0.6742644218780195, 'waste'
-0.5352673454310283, 'rate'
-0.5303849611788134, 'bland'
-0.42829568767759657, 'disappointed'
-0.40398760020368857, 'rating'
-0.3694813601780162, 'hi'
-0.3465757596972591, 'wrong'
-0.3306152227348906, 'bitter'

Table for Top 10 feature theta

4 RELATED LITERATURE

The dataset we use is from Kaggle (www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions). This data has previously been used to help users generate personalized recipes. The model is trained on the recipes and ratings the user has previously used.

For details, click this link.
https://aclanthology.org/D19-1613/
Generating Personalized Recipes from Historical User Preferences
Bodhisattwa Prasad Majumder, Shuyang Li*, Jianmo Ni, Julian McAuley*
EMNLP, 2019

In addition, our code refers to the content in Chapter 8 Workbook.

5 RESULT AND CONCLUSIONS

For the linear_model.Ridge function, we chose a different alpha(l). Finally, when alpha is 1000, MSE is the smallest.
The comparison results are as follows:
alpha = 0.01, validation MSE = 1.2496679508046218
alpha = 0.1, validation MSE = 1.2496676828565394
alpha = 1, validation MSE = 1.2496650067289181
alpha = 10, validation MSE = 1.249638578713635
alpha = 100, validation MSE = 1.2494056813723995
alpha = 1000, validation MSE = 1.2489943453525172
alpha = 10000, validation MSE = 1.2647839880212952

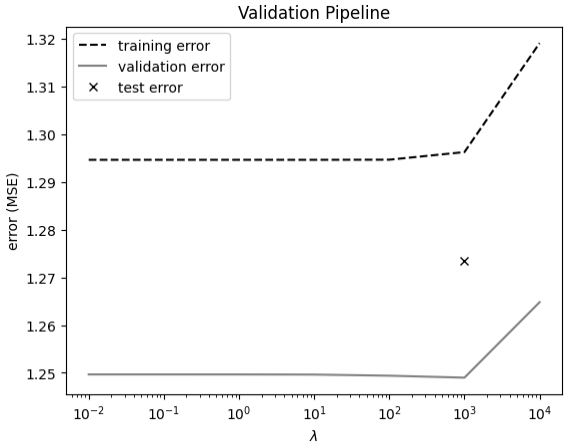

Figure 8 pipeline comparison

Finally we choose alpha =1000, validation MSE = 1.2489943453525172 . We use 0.3 as the threshold, and the difference between the predicted result and the real result is within 0.3, which is considered a successful prediction.

By comparing the number of successfully predicted data to the total number (in the test set), we get an accuracy of 79.22%. This result is within our acceptable range.