CSE 158 Assignment 2

November 27th, 2023

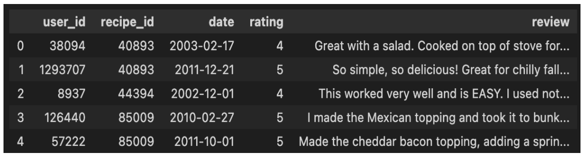
1. Dataset

Abstract

This assignment will present a predictive model that predicts a food recipe's rating by examining the review's language. The dataset of this predictive model can be found on food.com. The model is built upon sentiment analysis by using ridge regression from the linear model in scikit-learn. In other words, this model will collect all words from the dataset, and analyze the sentiment score of all words. For the rating of a recipe, The predicted rating will be the counts of each word multiplied by its sentiment score and the offset term.

The dataset consists of over 100,000 rows. Each row represents the interaction between a user and a recipe and has the metadata of recipe ID, user ID, date, rating, and review. The rating of each recipe would be in the range of 0 to 5. The figure below is an overview of a dataset.

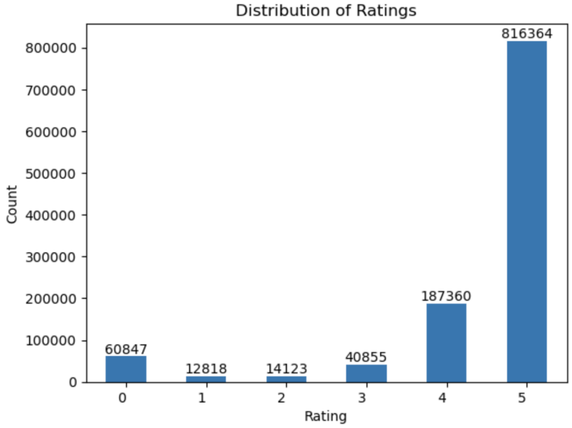
Overview of the Dataset



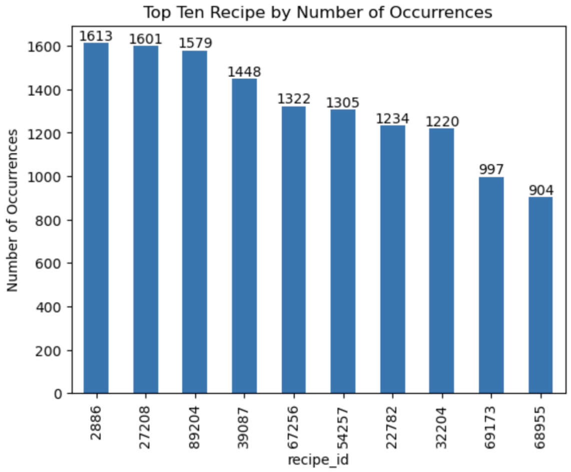
	user_id	recipe_id	date	rating	review
0	38094	40893	2003-02-17	4	Great with a salad. Cooked on top of stove for...
1	1293707	40893	2011-12-21	5	So simple, so delicious! Great for chilly fall...
2	8937	44394	2002-12-01	4	This worked very well and is EASY. I used not...
3	126440	85009	2010-02-27	5	I made the Mexican topping and took it to bunk...
4	57222	85009	2011-10-01	5	Made the cheddar bacon topping, adding a sprin...

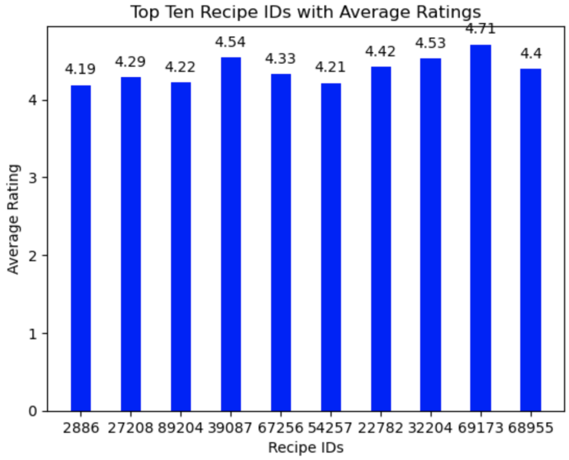
Exploratory Data Analysis

After performing some exploratory data analysis like the distribution of rating, the majority of recipes is between four and five. Zero rating surprisingly is more than one, two, and three ratings. The bar plot below shows the distribution of all ratings.

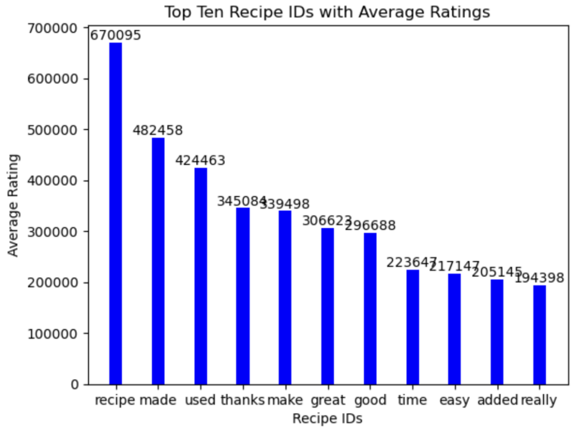


Besides the distribution of ratings, we also investigated the top ten recipes that have the most interaction, and their average rating. The figures below are "recipe_id vs Number of Occurance", and "recipe_id vs Average Rating". We can see that users tend to give higher ratings to the top ten popular recipes, which meets my expectations because 72% of all users give a rating of 5, and 16.5% of all users give a rating of 4. In other words, 88.5% of all users give a decent rating to the recipe, and the rest is the remaining.





However, it seems to be a problem for the model to perform decently since the data looks imbalanced. In other words, it would be more difficult for a model to distinguish between positive and negative reviews. Hence, we also include an overview of the positive sentiment score of top words. From the bar plot below, After filtering out stop words, “recipe” appeared the most, and other words appeared to be positive. Since most of the reviews are considered high ratings, it would be plausible that positive words occur more.



2. Predictive Task

Features and Label

After the broad overview of the dataset, we would predict the rating of a recipe based on the review of each (user ID, recipe ID) pair. The goal is to analyze how well the review captures the sentiment of users and predict the rating of the given recipe. To reduce insignificant features, we only examine the words that are top 1000 counts. Also, the length of the review would be an additional feature to increase the accuracy of the model. Moreover, features will be in the format of one-hot encoding. The dataset will be split into 90% for the training set, and 10% for the validation set. The evaluation metric of this model will be “mean square error (MSE)”.

$$MSE \ = \ \frac{1}{n}\sum^{n}_{i=1}(Y_{i} - Y^{'}_{i})$$

Baseline Model

The Baseline model would be predicting ratings that each recipe will take their own average. For validation, if the model encounters a recipe that has not been seen before, it will take the global average. The reason why we choose this model is that this model is straightforward and requires minimal computation.

Data Processing

For the review of the entire set, we decided to filter out all punctuations and stopwords because those words contribute insignificant effects on our prediction so that we could reduce irrelevant computation. Moreover, we use the algorithm of Porter Stemmer to transform every word in all reviews. In other words, it will remove the suffix of words that have similar meanings. For example, “working” and “worked” will be transformed into “work”. Finally, we remove reviews with null values.

3. Model

Given Each row in the dataset, The feature will be in 1000 dimensions plus the length of the review and the offset term. 1000 dimensions will be the most occurrences of the top 1000 words in the bag of words. The following equation shows the format of the features in one-hot encoding.

$$feat = [counts\ of\ popular\ word * 1000 , len(review), 1]$$

The following approaches that are discussed in the class would be used in our model:

- Bag of Words
- Sentiment Analysis
- Linear Regression

The reason why we choose these approaches is due to their efficiency and capability to capture the characteristics of each review. Hence, the model can be extracted into the following equation:

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

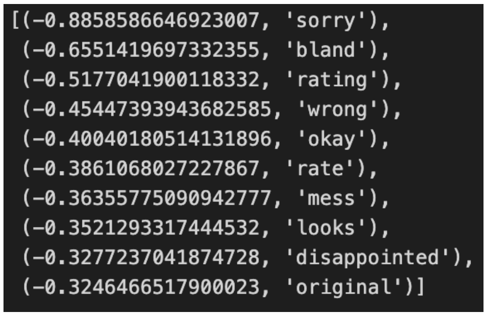
Description of Model Equation

Alpha is the constant term, theta is the sentiment score for each word, and text is the review and w is a word in a review. We only consider the words that are in the top 1000 words with the most occurrences. Hence, any words in a review that are not in the subset will be filtered out.

Overview of Sentiment Score

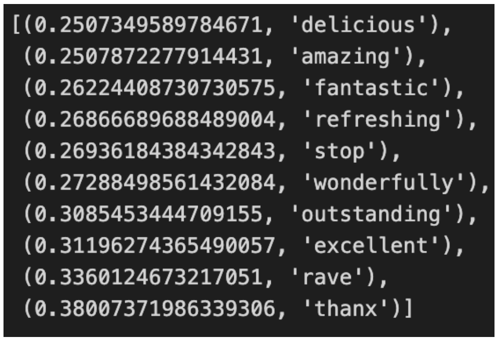
After training our model, we took a look at the sentiment score for the ten most negative and positive ten words:

Lowest Sentiment Scores



```
[(-0.8858586646923007, 'sorry'),
 (-0.6551419697332355, 'bland'),
 (-0.5177041900118332, 'rating'),
 (-0.45447393943682585, 'wrong'),
 (-0.40040180514131896, 'okay'),
 (-0.3861068027227867, 'rate'),
 (-0.36355775090942777, 'mess'),
 (-0.3521293317444532, 'looks'),
 (-0.3277237041874728, 'disappointed'),
 (-0.3246466517900023, 'original')]
```

highest Sentiment Scores



```
[(0.2507349589784671, 'delicious'),
 (0.2507872277914431, 'amazing'),
 (0.26224408730730575, 'fantastic'),
 (0.2686668968848904, 'refreshing'),
 (0.26936184384342843, 'stop'),
 (0.27288498561432084, 'wonderfully'),
 (0.3085453444709155, 'outstanding'),
 (0.31196274365490057, 'excellent'),
 (0.3360124673217051, 'rave'),
 (0.38007371986339306, 'thanx')]
```

As we expected, the top ten most negative and positive words in our training set sound plausible in their categories in general.

Baseline Model Performance

We performed the prediction of our baseline model, and here's the result of this model

$MSE\ =\ 1.5374058499993166$

Surprisingly, the baseline model performed well, which set a high benchmark for our model.

Model Performance

The Result of our model:

$MSE\ =\ 1.4314406101937072$

From comparing the MSE of each model, our model outperformed the baseline model.

Challenges

We had several unsuccessful attempts at the model because it could not beat our baseline model and had slightly greater MSE than our baseline model. Before we decided to choose the 1000 words with the most occurrences in the bag of words, we tried 100 and 500 words, and the MSE of each is about 1.72 and 1.60. After that, we realized that if we increased the quantity, the MSE of our model would decrease. Therefore, we keep increasing the quantity of the popular words to increase the performance of the model, but the MSE of our model does not reduce significantly and is stuck at around 1.45. The reason for this is that the words we increase will become more and more rare, which is less and less relevant to our prediction. It also increased the runtime of the model as we increased the size. Therefore, we decided to stop at 1000.

4. Literature

The Dataset we choose to work on is from Kaggle.com and produced by Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Also, it could be found in Julian McAuley's datasets on Recommender Systems and Personalization Datasets.

As we searched more on the internet, most of the previous works that were done based on the dataset we worked on were creating a personal recipe for users by using neural networking. They not only used the review and rating of a recipe, but also used most of the features in datasets like the ingredients, nutrition, and e.g. One paper that we found related to this dataset is *Generating Personalized Recipes from Historical User Preferences* written by Professors Julian McAuley and his researchers. It demonstrates how they created a personal recommendation for a user based on their history preferences, and the features of their favorite recipes by using neural networking. Compared to our model, their model would outperform ours if it performs the same predicting task as ours since they use more advanced machine learning algorithms that can be more flexible to any data.

State-of-the-art Methods

As far as we research, the most modern and advanced methods on this dataset we could find are neural networking and deep learning. These methods might be better performed than our model since they would create layers that can handle more complex features and predict the rating of recipes with higher accuracy.

5. Conclusion

The best MSE for our model is around 1.43 by using the methods of Bag of Word, Sentiment Analysis, and Linear Regression. We tried to increase the size of the most popular word for smaller MSE, but it not only didn’t reduce MSE significantly, but it increased the runtime of our model. Therefore, we believe that it’s the best choice to only examine the top 1000 words with the most occurrences. The parameters of our models can be described as alpha as y-intercept plus the sum of the words in a review and the top 1000 most occurrence bag of words multiplied by its sentiment score. We also include the length of the review as a feature to increase accuracy. As a result, the MSE beat our baseline model that predicts the rating of a recipe based on its average.

Citation

Dataset:

Food.com Recipes and Interactions, Julian McAuley, https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions

Literature:

Generating Personalized Recipes from Historical User Preferences, Bodhisattwa Prasad Majumder*, Shuyang Li*, Jianmo Ni, Julian McAuley EMNLP, 2019 https://www.aclweb.org/anthology/D19-1613/