Predicting Amazon Fresh Foods Ratings With Language Feature Analysis

Peter Liu
University of California: San Diego
p3liu@ucsd.edu

Onel Shina
University of California: San Diego
oshina@ucsd.edu

Margarita Leonova
University of California: San Diego
mleonova@ucsd.edu

Jaeyeong Hwang
University of California: San Diego
jah166@ucsd.edu

Abstract

In this CSE 158 assignment, our team investigated Amazon Fine Foods reviews to create a predictor for ratings. Through use of linear regression, we created a supervised learning model that demonstrated a correlation between a set of textual features and review ratings.

Our dataset consists of features such as the reviewer's rating, review text, review summary, helpful votes and unhelpful votes on the review, and the total number of votes associated with the review. Additionally, we discovered that we could apply language processing to obtain advanced features which provided us with excellent correlating features. Through use of optimizations such as n-grams, regularization, sentiment analysis, and rounding of edge cases, we managed to improve upon the baseline model.

We compare our model's results with the baseline model by using Mean Squared Error (MSE) as the metric. The baseline resulted in an MSE of 1.738211814923221 on the test set. Our model resulted in an MSE of 0.8608320287972824 on the test set, an improvement over the baseline by 50.475999449 percent.

Keywords

Linear Regression; Ridge Regression; Feature Analysis; Natural Language Processing; Bag-of-Words; N-Grams; Sentiment Analysis; Regularization

Introduction

Finding a reliable dataset to study is the crux of data analysis. Without a well documented and plentiful dataset, it can be difficult to identify patterns and relationships between aspects of the data. When identifying a dataset suitable for our team, we singled out specific datasets with characteristics conducive for study:

- Reasonably Large Size (50,000+ Samples)
- Relevant information fields available
- Open for different interpretations between different models

With these factors in mind, our team decided to investigate predictive relationships from Amazon's expansive dataset of fine foods reviews. With 568,454 total reviews aggregated across a period of over 10 years from Oct 1999 to Oct 2012, we had a high quality dataset with an extensive amount of data to train, validate, and test on. This dataset also captures a variety of critical information fields that could be applied in further analysis.

Our overall goal in this assignment is to improve upon our baseline predictor through extracting text features from this dataset. In particular, we aim to identify the importance of language processing techniques, such as text mining, on the accuracy of prediction.

1. Dataset

We utilized Amazon's Fresh Foods [6] dataset for this study. As of now, the Amazon dataset from Stanford is advertised with this schema and with these statistics:

Dataset Format

productId	An ID representing the product
userId	An ID representing the user
profileName	The name of the user
helpfulness	Rated Helpful / Total Times Rated
score	Rating of the review
time	Timestamp of when the review was made
summary	Short summary of review content
text	Full review text

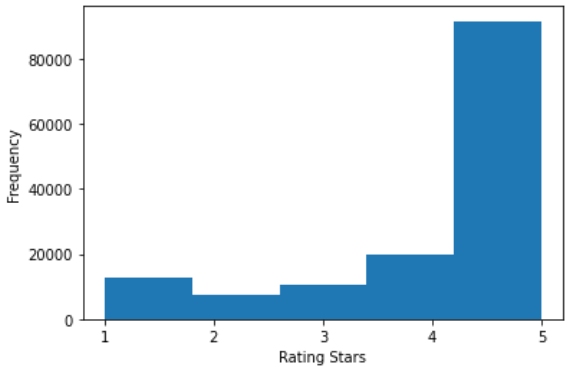
Dataset Statistics

Number of reviews	568,454
Number of users	256,059
Number of products	74,258
Users with > 50 reviews	260
Median no. of words per review	56
Timespan	Oct 1999 - Oct 2012

To get a better understanding of the dataset and the distribution of ratings, we decided to extract some important statistical information from the dataset and plot them.

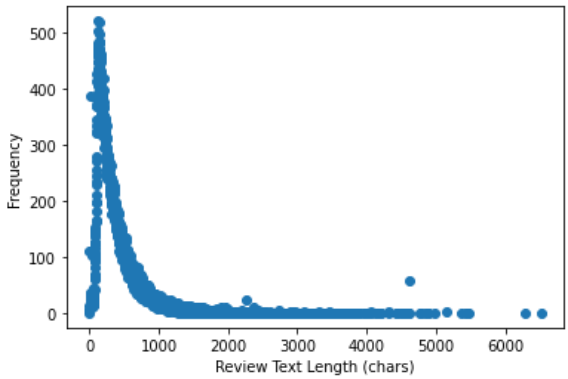
First, we looked at the distribution of ratings among the reviews, the mean rating among all training reviews was approximately 4.2 and the median rating was 5.0 with the following distribution histogram:

Figure 1: Distribution of Ratings



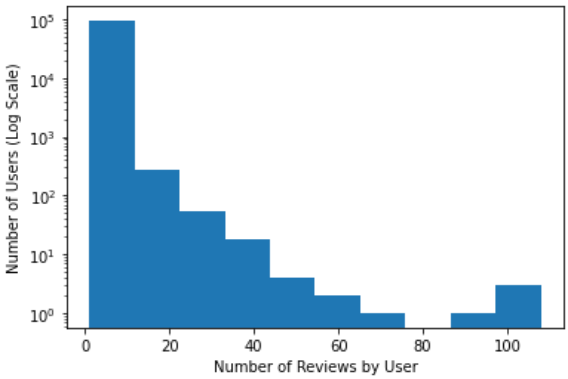
Then we decided to look at the distribution of text review length in characters in the training dataset. The mean review length in characters was approximately 383.5 characters, and the median length was 279 characters, with the following frequency plot:

Figure 2: Distribution of Review Text Length



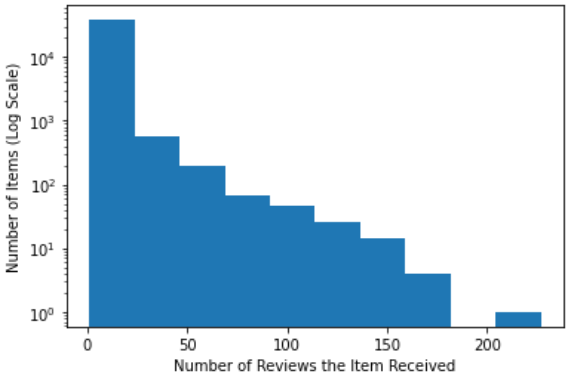
Moreover, we also examined the number of reviews per user. The mean number of reviews per user in the training dataset was approximately 1.47, and the median was 1 review per user. With the following distribution plot:

Figure 3: Distribution of Reviews per User



Also, we examined the number of reviews per item, and we found that the mean number of reviews per item was approximately 7.36 reviews, and the median was 2 reviews per item, With the following distribution plot:

Figure 4: Distribution of Reviews per Item



To further analyze the dataset, we examined the helpfulness of reviews, around 48% of the training data reviews did not have negative or positive helpfulness ratings, while around 52% of the training data reviews were rated once or more as helpful or not helpful. The average helpfulness of reviews in the training data was approximately 0.776.

2. Predictive Task

2.1 Overview

Our goal for this assignment was to predict what review score a person will give to a particular food based on the several data fields in the dataset. After extensive analysis of our dataset, we concluded that the features like review helpfulness, review score, review summary and review text, could all be relevant in predicting the review score. Therefore, we decided to consider these fields as features for building our model.

2.2 Features

For the text fields such as review text and review summary we decided to use the bag-of-words text processing technique to extract features. We examined review helpfulness to calculate the total amount of helpful and unhelpful reviews. Additionally, we considered review summary and review text length as features. We experimented with a combination of some of these features to determine which set of features gives us the best result.

In picking the best model, our goal was to compare how different models will perform and choose the one that gives us the lowest Mean Squared Error (MSE), which is defined by the following mathematical formula:

$$MSE = 1/N \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2$$

Our model will be generated through Linear Regression, which is described by the following formula:

$$Y = \sum_{features} \theta_f \times X_f$$

Linear regression is a linear model that attempts to minimize the sum of squared residuals between predicted and truth values. We use this model, because it can quickly identify correlations between specific features.

2.3 Evaluation

To accurately evaluate our model and avoid overfitting, we split our dataset into training, validation and test sets as follows:

- Training – 50% (220914)
- Validation – 25% (110457)
- Test – 25% (110457)

For training our model we use the training set, we use validation set to optimize our model and test the model on the test set. We will evaluate the result of our model against our baseline (mentioned later) by comparing how each model fares against the testing set. The better our model performs, the lower the MSE relative to the baseline.

2.4 Baselines

When identifying our baseline model our team came up with two options:

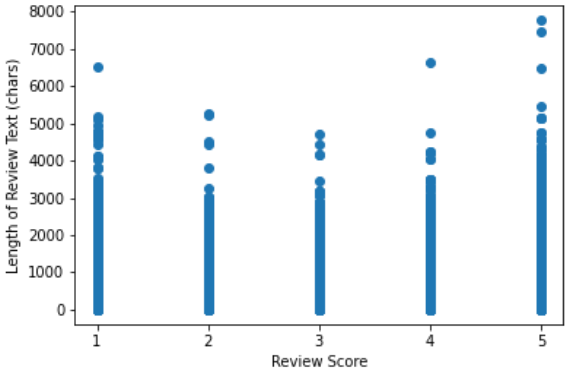
- use a simple predictor that predicts food rating based on the mean review score
- predict food rating based on the review length

For the first baseline, we calculated the mean review score using the training data, then we applied it to every data point in the testing data. This baseline gave us a Mean Squared Error of 1.738211814923221 on the testing set.

For the second baseline, we utilized a model built on linear regression to evaluate the relationship between the review text length and the review score. We trained this model on the training data and then we used it to predict the review scores of the testing data.

Overall, we decided to choose the first baseline over the second one. Although its Mean Squared Error of 1.7276612907564608 was actually lower than the Mean Squared Error of the mean review baseline, it was not a significant improvement either. A visual representation of the length of review score to the review text length is provided in Figure 4.

Figure 5: Distribution of Review Score per Length of Review Text



As we can see in Figure 5, there is not a significant correlation between the length of the review text and the actual review score.

Additionally, we decided to exclude the second baseline due to another practical reason. Because our goal of this assignment was to demonstrate exactly how utilizing features from a dataset could improve a basic model, we decided against using the length of review baseline as it inherently contains one of the possible features that we might explore for the project.

3. Model

3.1 Bag-of-words

In text mining, bag-of-words is a model generated from text which utilizes text-frequency in order to train a model. Our dataset provides textual information for both the review content and review summary. Through bag-of-words, we can interpret this textual data as values within a feature array. For each individual review, a distinct amount of word frequencies will appear, causing a unique variation on the feature values generated. Feature values which are generated from bag-of-words models vary based on frequency and weight of words, allowing for future statistics to be inferred for the validation and test dataset.

In our model, we used bag-of-words on both the review text and the review summary with the

top 100 most positive and negative sentiment words found from the 1000 most frequent words across all reviews in the training dataset.

3.1 Combining Basic Features With Bag-of-Words

Unnecessary features may add noise when training our predictor. When approaching our model, we decided to run an ablation experiment to determine exactly which features were relevant to our model. In this experiment, we incorporated:

1. Top 100 Popularity Bag-of-Words from Review Text
2. Review Text Length
3. Top 100 Popularity Bag-of-Words from Review Summary
4. Review Summary Length
5. Total Interactions (helpful and unhelpful)
6. Helpful Interactions
7. Unhelpful Interactions

As observed by Table 1, we noticed that the main features that impacted our MSE's accuracy were the top 100 popular review text words and the top 100 popular review summary words. Removing one of these two features increased the MSE of our result by a significant amount, thus demonstrating that those features were instrumental in prediction.

Table 1: Ablation Experiment

Model	Test MSE
All Features Enabled	1.2364637602642083
Top 100 Popularity Bag-of-Words from Review Text Excluded	1.4784152159533894
Review Text Length Excluded	1.236463760264209
Top 100 Popularity Bag-of-Words from Review Excluded	1.38593749865645
Review Summary Length Excluded	1.2364637602642086
Total Interactions Excluded	1.2364637602642086
Helpful Interactions Excluded	1.2364637602642086
Unhelpful Interactions Excluded	1.2364637602642086

3.2 N-grams Parsing (Pre-processing)

Although initial results demonstrated a correlation between the frequency of 1000 popular words to review rating, additional preprocessing was attempted to improve the accuracy of our results. Through use of a language technique, n-grams, we are able to further identify topics of interest. N-grams represent a set of words within a given window size. For example, if we were looking at window size n=2 (bigrams), we would be identifying specific pairs of words over individual words (unigrams). To identify the effectiveness of N-grams parsing, we utilize linear regression with three models in Table 2:

- The Unigrams model demonstrates our original bag-of-words model, utilizing only unigrams.
- The Bigrams model demonstrates a new bag-of-words model, which utilizes only bigrams.
- The Unigrams and Bigrams model demonstrates a combination of the first and second models, by incorporating a combination of the most popular from both

unigrams and bigrams.

Table 2 shows a reduced accuracy from Model 1 to Model 2 and from Model 1 to Model 3. With this knowledge, we recognize that utilizing purely unigrams provides the best accuracy for our model. In the case of utilizing n-grams, we massively increase the number of features, so we might be double-counting pre-existing features. Despite this, running models on n-grams lets us investigate our next topic: sentiment analysis.

Table 2: MSE of N-gram Parsing

Model	Train MSE	Validation MSE	Test MSE
Unigrams	1.06652159 0180318	1.09535226 8638909	1.08249321 97608654
Bigrams	1.22580861 27248826	1.24481240 77327882	1.24921608 07644945
Unigrams and Bigrams	1.10445194 07462328	1.13128492 72369578	1.11896316 70151514

3.3 Sentiment Analysis (Pre-processing)

While n-grams parsing alone may not generate substantial improvements towards our model, it does let us identify the most important features from our model. Utilizing the unigrams model which performed the best on the previous test, we can obtain the thetas from the linear regression model in order to identify the most positive and most negative unigrams (demonstrated in Table 3 and Table 4). We then obtain the top 100 most positive and negative unigrams and utilize these features as the exclusive features for our feature array. Poorly selected features may add noise during the training phase of our predictor, so we filtered out the features that generally did not have a significant impact on the result.

Figure 6: Most Positive Unigrams

Table 3: 5 Most Positive Unigram Sentiment

Positive Unigrams	Theta
pleased	0.36187356971504525
hooked	0.35813171145083833
beat	0.34808942593142
awesome	0.3452045085725943
satisfied	0.3367320685692052

Figure 7: Most Negative Unigrams

Table 4: 5 Most Negative Unigram Sentiment

Negative Unigrams	Theta
awful	-0.8506624657834093
terrible	-0.7945820379782581
horrible	-0.7928052315776569
return	-0.6359122606676837
disappointed	-0.5659952582749989

Overall, specifically targeting the most positive and negative unigrams gave us significantly better results than purely targeting unigrams. As seen in Table 5, the new model with sentiment analysis is a vast improvement from our older model based on popularity.

Ideally, we should be able to test on even more words and generate top 1000 or more positive and negative unigrams from top 10000 popular words (or more), but our machine was not able to procure a sufficient amount of memory to parse the data without crashing.

Table 5: Top 200 Popular Unigrams Against Top 100 Positive and Negative Unigrams

Model	Train MSE	Validation MSE	Test MSE
Top 200 Popular Unigrams	1.3220192627153153	1.3374376780449913	1.3297725042005872
Top 100 Positive and Negative Unigrams	1.1484330137 39955	1.1737946183 16245	1.1644870705603934

3.4 Incorporating Review Summary Through Bag-of-Words

Our earlier models were purely focused on the text of the review, but a review's text was not the only feature that we could run language analysis on. We also observed that our dataset contained a review summary, which was a short blurb of text aimed at introducing a topic of discussion for the review. This feature is short and concise, so a unigram bag-of-words model can provide a significant amount of insight into predicting a review's sentiment. To identify the effectiveness of incorporating review summary data, we utilized the same unigram sentiment parsing method above to create three models in Table 6:

- The Review Text model demonstrates using bag-of-words on only the review content
- The Review Summary model demonstrates using bag-of-words on only the review summary
- The Review Text and Review Summary model demonstrates using bag-of-words on both the review content and the review summary

Table 6: MSE of Review Text and/or Summary Bag-of-Words Models

Model	Train MSE	Validation MSE	Test MSE
Review Text	1.1484330137 39955	1.1737946183 16245	1.1644870705603934
Review Summary	1.0838806790803213	1.0970519799513556	1.0970493380687834
Review Text and Review Summary	0.8829975830794127	0.9015895336341492	0.8952940661849451

3.5 Regularization

Normal regression gives you unbiased regression coefficients. Utilizing ridge regression allows you to regularize coefficients. As the estimated coefficients are pushed towards 0, they work better on new data-sets. Regularization allows us to interpret complex models without overfitting at the same time. In ridge regression, we must set an alpha which defines the strength of the regularization performed according to this formula:

$$min\theta \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2 + \lambda||\theta||_2^2$$

In our model, we identified if our model was suffering from over-complexity by comparing the validation MSE values against multiple tuning values. These values are documented in

Table 7. Overall, the regularization parameter of 1 performed the best.

Table 7: Regularization Constant

Regularization Constant	Validation MSE
0.01	0.9015990585286275
0.1	0.9015979249228075
1	0.9015895336341492
10	0.9017615333965223
100	0.9127789457201247
1000	0.9928541131826174

3.6 Rounding Edge Cases

We noticed that some of our predictions are less than 1.0 or greater than 5.0, which lies outside the range of possible ratings. In the case that there are ratings above the max, we will cap them to the maximum. If there are ratings under the min, we will also cap those to the minimum. Additionally, since ratings must be an integer, we experimented with rounding, as some of our predictions were in between two integers. Capping implementation resulted in small increases to accuracy documented in Table 8.

Table 8: Rounding and Capping

Model	Train MSE	Validation MSE	Test MSE
Unchanged	0.8829975830794127	0.9015895336341492	0.8952940661849451
Capping	0.8480305811078144	0.8666464596108248	0.8608320287972824
Rounding	0.9748997347384405	0.9969399856957911	0.9941334637008067
Capping and Rounding	0.9388947735317816	0.9602831871225906	0.957938383262265

4. Literature

4.1 Description of Dataset

The dataset that we used for this assignment comes from the Stanford Large Network Dataset collection. The dataset was obtained by J. McAuley and J. Leskovec from public sources on the web and is a subset of a larger Amazon Reviews set that also includes other categories. The data is a collection of user reviews of Amazon fine foods that span a period of more than 10 years. Reviews include product and user information, ratings and text review. The dataset was used in a paper “From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews” where McAuley and Leskovec studied how to capture user taste evolution in product rating systems.

4.2 Similar Literature

There have been numerous studies done involving the Amazon Product Review dataset. Zheng et al. [1] aimed at predicting review scores on the same dataset we used in this study, they also found that using text features in regression models beat using other features that do not involve text.

Huang et al. [2] aimed at predicting review scores on RateBeer reviews dataset from SNAP [3] [4] [5]. They utilized the brewer ID and the review text to make their predictions. By combining Ridge, GBRT, and Random Forest models they achieved better results than using single models to make predictions. Unfortunately, our data did not have a datafield that reveals the item manufacturer as we believe such datafield can improve performance. Having information on the average score that a brewer receives for their beers can be helpful in predicting their future beer scores. Likewise, if this dataset contained manufacturer IDs for different food items, we could have utilized that information to make more accurate predictions on future food item scores for that manufacturer.

4.3 State-of-the-art methods

One of the biggest challenges of working with text reviews is that unlike ratings, they are not quantitative. Reviews can vary in length, have different syntax and include many words that are not useful for the analysis. Natural Language Processing techniques make it easier and faster to analyze customer reviews and build a recommender system based on the feedback.

By analyzing several studies on Amazon Review Dataset we came to a conclusion that the most common text preprocessing techniques include:

- Normalization - removing punctuation
- Tokenization - splitting review text into a list of tokens
- Stemming - reducing the word to its stem (root form)
- Lemmatization - reducing the word to its dictionary form
- Removing stop words - these are the most common words in a language

The most common feature extraction techniques are:

- Bag-of-Words - describes the occurrence of words within a document
- TF-IDF - evaluates how relevant a word is to a document in a collection of documents
- Word2Vec - technique to learn word associations from large corpus of text

These findings helped us feel confident about our approach in obtaining the most efficient model. Just like in other studies, we used a combination of text preprocessing techniques, sentiment analysis, and bag-of-words to build our feature vector.

5. Conclusion

After evaluating two baselines for this predictive task, we established that using mean scores to predict the review scores was the baseline more aligned to our goals within this project. When examining features available in our dataset through an ablation experiment, we noticed that review text length, review summary length, and positive and negative votes on the review all did not provide a relevant improvement. When looking at features generated through language processing, however, we noticed that our results were significantly better than the baseline. Due to the previous ablation experiment, we observed that adding other features does not improve our model's accuracy, and thus we made an executive decision to exclude them from our final model to lower the chance of overfitting.

It is a common phenomenon that when our training MSE decreases, our validation and test MSE will decrease as well. We acknowledge that there may be a possibility of overfitting on the training data, but our results demonstrate that our model is able to perform consistently on all the datasets: training, validation, and testing.

Following the previous model, we improved our model by utilizing a variety of pre and post processing techniques. We implemented preprocessing techniques through generating unigrams and identifying which unigrams carried the most positive and negative sentiment. Through this sentiment analysis, we were able to identify target words that had relevant correlations to our model. This form of sentiment analysis was utilized to construct later bag-of-words models and provided a significant gain in accuracy of predictions. For post processing, we implemented capping of the edge cases. In a few predictions, we noticed that some of the results were over 5 and under 1, which was out of the boundaries for possible ratings. For these cases, we limited the result to be within the boundaries specified. This form of postprocessing was also able to provide us with marginal improvements. With these techniques, we documented our final results in Table 9. Overall, we were able to improve over the baseline model by %50.475999449.

While conducting our experiment, we noticed that our machine ran out of memory when attempting to try larger sentiment analysis models. These models contained dictionaries of 2000 or more words. Future room for exploration would involve obtaining a stronger machine, preferably with 64 gigabytes of RAM or higher. Then we would attempt to run models obtaining the top 1000+ most positive and negative sentiment choice words from the top 10000 popular words. Demonstrated by previous trends, this would most likely improve our model accuracy by a significant margin.

Beyond hardware limitations of our project, there are also many other language processing techniques that may be examined to improve our model. One example would be stemming, which

merges different inflections of words and can help model a more accurate distribution. By reducing complex inflections into their root form, we can often combine features that would otherwise represent the same meaning. This may have increased the accuracy of our model, but due to time constraints we were unable to implement the technique.

Table 9: Final Results with our Model: 100 Most Positive & Negative Bag-of-Words for Review Summary and Text + Capping Edge Cases

Model	Test MSE
Baseline	1.738211814923221
Final	0.8608320287972824

6. References

[1] C. Zheng, Y. Zhang, and Y. Huang. Rating prediction on Amazon Fine Foods Reviews.

[2] J. Huang, S. Chen, Z. Gu, and K. Yang. Using Regression Methods to Predict Alcohol Beer Rating

[3] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[4] J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In Data Mining (ICDM), 2012 IEEE 12th International Conference on, pages 1020–1025. IEEE, 2012.

[5] J. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In Proceedings of the 22nd international conference on World Wide Web, pages 897–908. International World Wide Web Conferences Steering Committee, 2013.

[6] J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.