Amazon Mini Project Report

Zhiyu Lei

***Abstract*— This report provides an overview of the Amazon Fine Food Reviews dataset, including descriptive analyses and predictive analyses. Some text mining procedures, dimension reduction methods, and linear regression models were explored to predict review scores. A model with the RMSE of 1.0936 was obtained at last, which can be used to predict for new reviews.**

I. INTRODUCTION

The dataset consists of 568,454 reviews of fine foods from Amazon, spanning a period of more than 10 years from October 1999 to October 2012. To find the patterns and insights in this dataset and then to predict review scores based on other attributes, some descriptive analyses were first conducted, showing the distribution of scores is highly imbalanced and the scores had different patterns over time. Then, some text processing methods, TF-IDF vectorization as well as Principal Component Analysis were applied to extract and select features from the text in order to feed into the linear regression models. Finally, different weighting methods on the training samples and different types of linear regression models were explored and evaluated, with one model good at predicting low scores and another good at predicting high scores.

II. METHODS

A. Data Preprocessing

Since the *Time* column in raw data stores the data of review in terms of seconds (the 0^{th} second is the first second of the year 1970), which is difficult to interpret and aggregate, it was transformed to store real date value in "YYYY-MM-DD" format. Also, the *Summary* column has missing values, which should imply the corresponding reviews do not have summaries, so the missing values were filled with empty strings. The HTML tags were removed as they are not necessary for review analysis. Finally, there are a few reviews having *Helpfulness Numerator* greater than *Helpfulness Denominator*, which makes no sense, so such numerator values were replaced with denominator values for simplicity.

Then after data cleaning, three new columns were created: *Summary Length*, *Text Length* and *Helpfulness Ratio* (for those ratings with both *Helpfulness Numerator* and *Helpfulness Denominator* being zero, define *Helpfulness Ratio* to be zero), and some basic descriptive analyses were also run. Meanwhile, a Latent Dirichlet Allocation (LDA) analysis was applied to the review texts to extract five potential topics in the reviews.

Next, the texts were processed in the following steps: 1) join the summary and main body together; 2) convert to lower cases and remove non-alphabetic characters; 3) tokenize the text with the Punkt sentence tokenizer; 4) remove English stopwords and reduce tokens to their root forms with the WordNet Lemmatizer.

B. Prediction Model

The entire dataset was split into two parts: a training set containing 468,454 reviews to train the models to predict scores, and a testing set containing 100,000 reviews to test the model performances.

A TF-IDF vectorizer was first trained to extract the text features and convert the texts into vectors for model input. Only 300 unigrams and bigrams with the highest term frequencies are extracted as features. Then, a Principal Component Analysis model was trained to reduce the dimensionality of the vector space with explaining at least 80% of the variance.

Some more features were also added. Since the line graph of average *Score* aggregated by day over time (Figure 1) shows a great fluctuation before 2007 and a slight decreasing trend after 2007, a new feature is created with value 0 if the review is before 2007, and with number of years after 2007 otherwise. As for *Summary Length* and *Text Length*, since these two attributes are right-skewed as suggested by histograms in Figure 2, they were transformed by taking natural log to reduce skewness (one was added to the original *Summary Length* to prevent computing $\log 0$).

The reviews in the training set were weighted to train linear regression models based on the features created previously. The weights were according to the helpfulness information. Basically, the more people found the review helpful and the higher the helpfulness ratio is, the higher quality the review should have. Thus, the weights were defined with:

$$weight = \begin{cases} numerator^{ratio} & \text{if } numerator > 0 \\ 1 & \text{if } numerator = 0 \end{cases}.$$

An alternative weighting method was based on the inverse ratio of the target frequencies as the dataset is highly imbalanced with far more reviews scored higher than those scored lower.

Two simple multiple linear regression models were first trained with two weighting methods respectively. As the review scores are only integers between 1 and 5, the model outputs are rounded to the nearest integer, and any outputs less than 1 are fixed to 1 while any outputs more than 5 are fixed to 5. The models were then evaluated with root-mean squared error (RMSE) on the testing set. Another regression model with unweighted samples was also trained for comparison. Since there were quite many features included in

the linear regression model which might lead to overfitting, a Ridge regression and Lasso regression model were tried as well to punish high values in the model coefficients.

III. RESULTS

A. Descriptive Analysis

Statistics including minimums, averages, medians, and maximums for text length, summary length, score, and helpfulness ratio are contained in Table I.

TABLE I
DESCRIPTIVE STATISTICS

	Text Length	Summary Length	Score	Helpful Ratio
min	3	0	1	0
avg	79.0964	4.1131	4.1832	0.4079
med	56	4	5	0
max	3377	42	5	1

The line graphs of average text length, summary length, score, and helpfulness ratio aggregated by day over time are contained in Figure 1. There were very great fluctuations in average review length, summary length, score, and helpfulness ratio from day to day between 2004 and 2007. After 2007, average review length and summary length remained quite stable, while average score slightly reduced, and average helpfulness ratio fairly dropped, especially after 2010.

The LDA analysis performed on the review texts in an unsupervised manner suggests five potential topics with the following top 10 keywords:

1) *food, dog, like, eat, dogs, treats, love, loves, just, good*. This topic seems to represent good reviews on dog food.
2) *tea, flavor, like, coffee, taste, good, chocolate, just, cup, drink*. This topic seems to represent good reviews on drinks, especially tea.
3) *like, good, taste, great, just, flavor, love, chips, salt, really*. This topic seems to represent good reviews on chips.
4) *coffee, amazon, product, price, good, great, order, just, buy, box*. This topic seems too represent good reviews on coffee products.
5) *product, water, like, sugar, use, taste, oil, just, good, bottle*. This topic seems to represent good reviews on cooking ingredient such as water, sugar and oil.

B. Prediction Model

Before training of the linear regression models to predict scores, a PCA model was trained to reduce the dimensionality of the feature space resulted from TF-IDF vectorization. Exploring the first principal component, the top 5 text features positively contributing to it are "*coffee, tea, cup, flavor, taste*," and the top 5 text features negatively contributing to it are "*love, cat, treat, food, dog*." Many features are also included in the topic keywords resulted from the previous LDA analysis, so we can conclude the PCA does make sense and is effective.

With the feature related to time and the log-transformed *Summary Length* and *Text Length* added, the first simple linear regression model was trained with samples weighted based on helpfulness, obtaining the accuracy of 0.4224 and the RMSE of 1.1008 on the testing set. The corresponding confusion matrix is shown in Table II.

TABLE II
CONFUSION MATRIX FOR FIRST REGRESSION MODEL

	1	2	3	4	5
1	185	1404	4771	2844	143
2	38	463	2323	2194	185
3	20	356	2639	3802	515
4	10	136	2313	8271	3463
5	9	190	4133	28914	30679

The alternative weighting method based on inverse ratio of target frequencies turned out to be even worse, with the accuracy of 0.2317 and the RMSE of 1.3387. The corresponding confusion matrix is shown in Table III.

TABLE III
CONFUSION MATRIX FOR SECOND REGRESSION MODEL

	1	2	3	4	5
1	958	4747	3337	285	20
2	213	2054	2570	344	22
3	118	1851	4205	1064	94
4	28	1225	6963	5030	947
5	63	2367	21389	29182	10924

The model trained with unweighted samples surprisingly had the best performance on the testing set, with the accuracy of 0.4262 and the RMSE of 1.0936. The corresponding confusion matrix is shown in Table IV.

TABLE IV
CONFUSION MATRIX FOR THIRD REGRESSION MODEL

	1	2	3	4	5
1	133	1199	4767	3109	139
2	32	376	2385	2268	142
3	11	293	2583	3983	462
4	5	108	2096	8672	3313
5	7	113	3410	29535	30860

Exploring the last two confusion matrices, we can find the model trained with samples weighted based on inverse ratio of target frequencies is good at predicting low scores while the model trained with unweighted samples is good at predicting high scores. Thus, if ensemble methods were allowed, an ensemble model with these two as base models could be trained to take the advantages of both.

Finally, in order to prevent the potential issue of overfitting, a Ridge regression and Lasso regression were also tried. Unfortunatly, punishing high values in the model coefficients has no help for increasing the predictive power. Even a very low regularization constant α would decrease the model performance.

APPENDIX

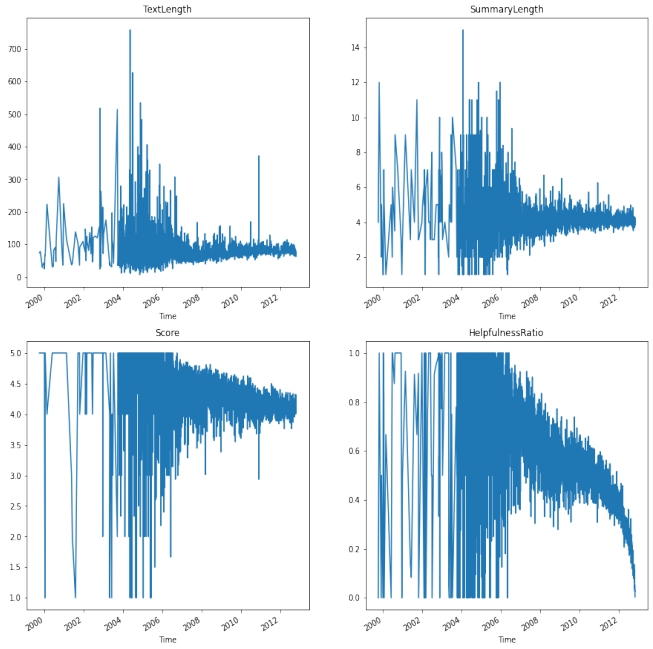

Fig. 1. Line graphs of variables over time

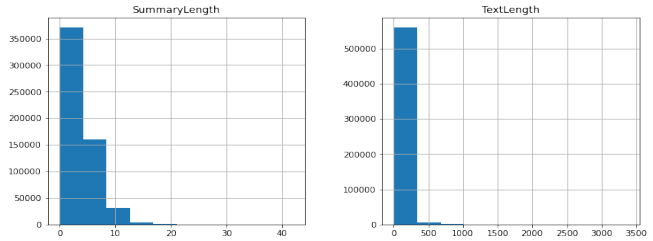

Fig. 2. Histograms of summary and text lengths

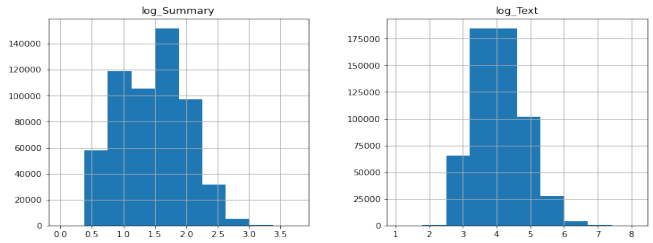

Fig. 3. Histograms of log-transformed summary and text lengths

REFERENCES

[1] S. Kapadia, "Topic Modeling in Python: Latent Dirichlet Allocation (LDA)" in Towards Data Science.

[2] Y. Berdugo, "Review Rating Prediction: A Combined Approach" in Towards Data Science.