Data Mining and Predictive Modeling of Amazon Customer Reviews

COSC 757.101 – Data Mining: Midterm Progress Report

W. Grant Hatcher, Kevin McNamara
Department of Computer and Information Sciences, Department of Marketing
Towson University, Maryland, USA 21252, USA
Emails: whatch2@students.towson.edu, kmcnam3@students.towson.edu

***Abstract*—In this proposal, we explore a set of Amazon customer reviews to analyze patterns between various attributes and develop a predictive tool for discovered correlated variables. Considering a dataset of over 130 million reviews provided from Amazon's AWS service, we have the potential to consider the impacts of positive and negative reviews on product ratings and customer sentiment. Moreover, applying advanced machine learning systems, such as deep neural networks in TensorFlow, we have the potential to provide high-accuracy predictive capabilities to be leveraged for consumer targeting, increased customer satisfaction, and the generation of additional revenue.**

***Index Terms*—Data mining, Machine learning, Exploratory Data Analysis**

I. Introduction

Since Amazon's launch in the mid-90's, their customer base have left hundreds of millions of reviews on purchased items. Through AWS, Amazon has availed over 130 million reviews in TSV files that are organized by product categories. With the 43.5% market share in 2017, Amazon is the leader in online retail in the US. The tech giant possesses a wealth of data regarding customer purchases, preferences, reviews, and more. Using data mining techniques, we can harness this data to research patterns in customer sentiments and buying habits.

Our initial analysis of the customer review data shows some uniformity across product segments. We are attempting to use the machine learning to predict Star Rating and Helpful Votes for individual product purchase reviews. The distribution of star rating is very similar from category to category. This presents potential challenges in finding a highly-accurate algorithm to predict the star rating. In addition, the helpful vote fields have some regularity as well.

II. Approach

A. Data Description

The dataset we are evaluating is the Amazon Customer Reviews Dataset available from Amazon Web Services (AWS) cloud. The dataset consists of more than 130 million reviews of Amazon products, separated into 46 subcategories (apparel, books, grocery, jewelry, luggage, etc.), each with over a million reviews. The dataset includes the attributes *customer_id*, *helpful_votes*, *marketplace*, *product_category*, *product_id*, *product_parent*, *product_title*, *review_body*, *review_date*, *review_headline*, *review_id*, *star_rating*, *total_votes*, *verified_purchase*, and *vine*. More specifically, the field *marketplace* is always "US" for United States, and the field *product_category* is uniform within each subcategory file (apparel, books, etc.). In addition, *helpful_votes*, *total_votes*, and *product_parent* are integer values, and *review_date* is a date of format YYYY-MM-DD. The attributes *product_title*, *review_body*, and *review_headline* are all variable-length strings. Finally, *product_id* and *review_id* are alphanumeric strings of capital letters and numbers of varying length.

B. Accessing the Dataset

First, we needed to access the data from AWS. This proved to be challenging as the files were quite large in size and were difficult to load in together. With over 130 million reviews, we needed to use a server with enough memory and processing power to load and manipulate such a large set of data. This was accomplished using a Dell PowerEdge R910 server with Intel Xeon E7530 processor and 64GB memory running Ubuntu Linux 16.04.5 LTS. Each data file, in .tsv or tab-separated file format, was downloaded via URL link. The ability to read in these data files was difficult, as most traditional softwares, especially on Windows PCs, could not open the file, and would significantly slow down usage when attempting any edits.

C. Data Preprocessing

Another challenging aspect of the process was preparing the data for analysis. In such a large dataset the amount of missing data and anomalies is quite substantial, and with individual files on the order of several gigabytes, it is impossible to individually search through the rows of data instances. Getting this data normalized and prepared for the next phase of the process was certainly a challenge. In this phase, it became evident that analyzing the entire dataset (broken up into 43 product category sets in 46 files) would be nearly impossible without careful consideration of programming operations. For instance, simple processes were taking significant amounts of time and some just simply would not work on the entire set even on individual product category segments of data (individual files). As an example, trying to loop even once through several million data items is not feasible.

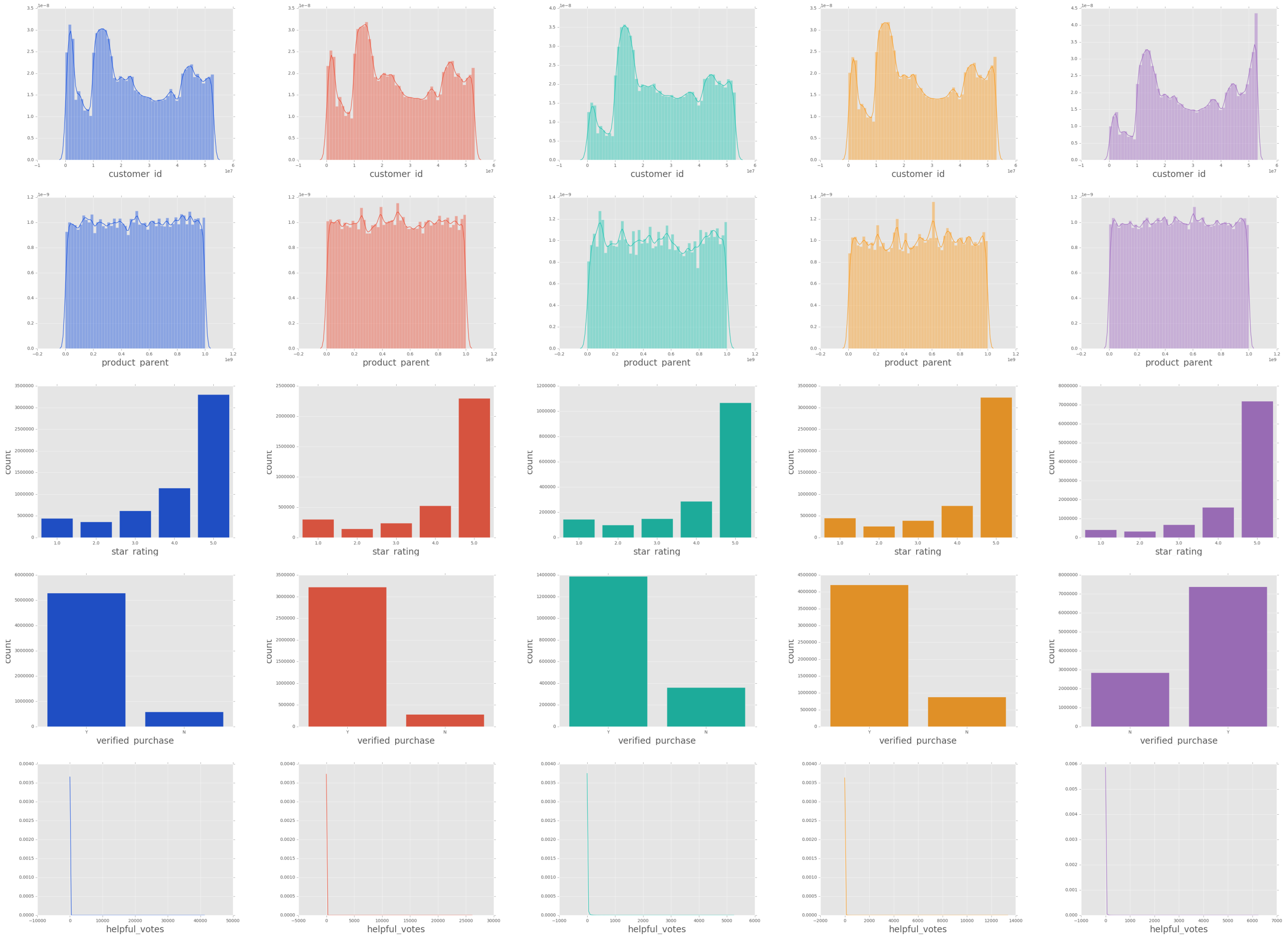
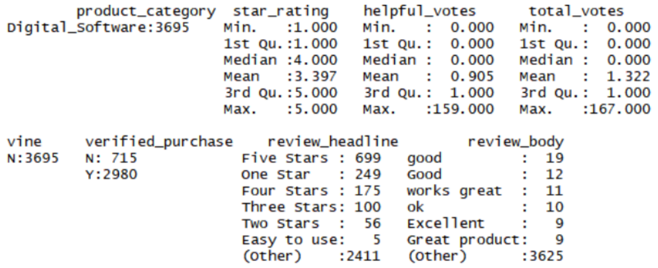


Fig. 1. Histograms and Density Estimation plots of product categories Apparel, Automotive, Baby, Beauty, and Books (left to right). Attributes are *customer_id*, *product_parent*, *star_rating*, *verified_purchase*, and *helpful_votes* (top to bottom).

To overcome the problem of assessing the dataset, the data was loaded into Python as a DataFrame, where operations can be performed on the entire matrix at once. This includes dropping NaN and missing characters, rows with too many columns, and values that do not match, especially for categorical variables. In this case, we are using Python 2.7.

D. Exploratory Data Analysis

Moving forward into our EDA phase, we limited our data evaluation to a subset of the entire dataset. Thus, we have analyzed and cross-referenced 7 different product category segments. These segments are: Apparel, Automotive, Baby, Beauty, Books (part 1 of 3), Camera, and Digital Software. Grant has successfully loaded and preprocessed the first three segments (Apparel, Automotive, Baby, Beauty, and Books) into his server using PowerShell. Kevin has successfully loaded the 6th and 7th segments (Camera and Digital Software, respectively) into R Studio. Initial EDA details for the Camera and Digital Software segments are shown in Figure 2 and Figure 3, respectively.



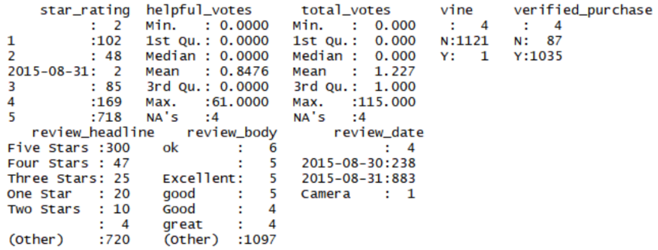
```
         product_category  star_rating    helpful_votes      total_votes
Digital_Software:3695    Min.   :1.000   Min.   :  0.000   Min.   :  0.000
                         1st Qu.:1.000   1st Qu.:  0.000   1st Qu.:  0.000
                         Median :4.000   Median :  0.000   Median :  0.000
                         Mean   :3.397   Mean   :  0.905   Mean   :  1.322
                         3rd Qu.:5.000   3rd Qu.:  1.000   3rd Qu.:  1.000
                         Max.   :5.000   Max.   :159.000   Max.   :167.000

 vine      verified_purchase     review_headline          review_body
 N:3695    N: 715             Five Stars : 699   good           :  19
           Y:2980             One Star   : 249   Good           :  12
                              Four Stars : 175   works great    :  11
                              Three Stars: 100   ok             :  10
                              Two Stars  :  56   Excellent      :   9
                              Easy to use:   5   Great product:    9
                              (Other)    :2411   (Other)        :3625
```

Fig. 2. Data description of Camera category. Includes mean, median, mode, and quantiles of numeric data columns (*star_rating*, *helpful_votes*, and *total_votes*), as well as counts for binary attributes (*vine* and *verified_purchase*).

Looking at the histograms in Figure 1, we can see near identical distributions of customer id numbers connected to reviews. In addition, we can see that the product parents are also quite uniformly distributed. Moreover, looking at the start ratings of reviews, we see a significant number of five star reviews, much higher than any other rating, and generally

TABLE I. Averages of all seven category data descriptions.

AVERAGE	**customer_id**	**product_parent**	**star_rating**	**helpful_votes**	**total_votes**
Count	28371981	28371981	28371981	28371981	28371981
Average	4053140	4053140	4053140	4053140	4053140
Mean	27550919	507045022	4	2	2
Std. Dev.	15383160	288812824	1	19	20
Min	10043	31422	1	0	0
25% Quartile	14223243	256026516	4	0	0
50% Quartile (Median)	26304429	518488891	5	0	0
75% Quartile	41963264	760629282	5	1	1
Max	53096535	999983463	5	14679	14952



```
     star_rating  helpful_votes      total_votes      vine     verified_purchase
           :  2   Min.   : 0.0000   Min.   :  0.000    :   4    :   4
1          :102   1st Qu.: 0.0000   1st Qu.:  0.000   N:1121   N:  87
2          : 48   Median : 0.0000   Median :  0.000   Y:   1   Y:1035
2015-08-31:  2   Mean   : 0.8476   Mean   :  1.227
3          : 85   3rd Qu.: 0.0000   3rd Qu.:  1.000
4          :169   Max.   :61.0000   Max.   :115.000
5          :718   NA's   :4         NA's   :4
   review_headline    review_body          review_date
Five Stars :300    ok         :   6              :  4
Four Stars : 47               :   5    2015-08-30:238
Three Stars: 25    Excellent:   5    2015-08-31:883
One Star   : 20    good       :   5    Camera    :  1
Two Stars  : 10    Good       :   4
           :  4    great      :   4
(Other)    :720    (Other)    :1097
```

Fig. 3. Data description of Digital Software category. Includes mean, median, mode, and quantiles of numeric data columns (*star_rating*, *helpful_votes*, and *total_votes*), as well as counts for binary attributes (*vine* and *verified_purchase*).

exceeding the volume of all other star ratings combined. Traditionally, we see that nearly all reviews are verified purchases, and that the number of helpful votes are generally quite low, with only very few products getting many helpful review votes.

Considering all seven product categories, we can see in Table I the total number of samples considered to be over 28 million, with the average per category being only 4 million. In addition, we see that the vast majority of products have 0 total or helpful votes, with 1 vote being the value of the 75th percentiles. In contrast, we see that the vast majority of reviews have a star rating of 5, with the 25th percentile being a star rating of 4. Note that the attribute *star_rating* represents a categorical integer of 1 through 5, while *helpful_votes* and *total_votes* are unbounded integers.

E. Shallow Learning

Now that our EDA phase is complete we have initiated steps to apply some basic machine learning techniques (decision trees, logistic regression, etc.) to fit our data. However, this process is ongoing and not complete as of yet. Currently, for categorical prediction, we must consider the input space of all our diverse attributes. Our goal in the short term is to demonstrate class prediction on *star_rating* based on the remaining information, as well as prediction of *helpful_votes*, which necessitates regression analysis.

F. Modeling

As the long-term goal of our project, we are interested in prediction the helpfulness of a review based on natural language analysis. This requires natural language model building and sentiment analysis of the *review_headline*, *review_body*, and *product_title* attributes before integration with our more basic predictive models for *helpful_votes* and *star_rating*. To accomplish this, we hope to build a sentiment analysis neural network or use some other existing framework. If necessary, we shall implement parts of pre-built models.

III. Literature Review

In this section we review some relevant works to our current research project. Specifically, Diaz and Ng [2] provided an overview of relevant works on making predictions of helpful reviews. They stress the importance of context in understand the reviews. Also, they mention a lack of uniformity among approaches for predicting helpfulness which hindered their ability to compare methods. That being said, the authors specifically mention a few advance models such as probabilistic matrix factorization and HMM-LDA as well as neural networks as exciting prospects for predicting customer reviews.

In addition, Martin [1] in her 2017 unpublished masters thesis explored review text analysis in predicting review ratings. She cites differing user standards as a major hindrance to this method along with anecdotal information and differing vocabulary that users may use. Martin looked at two different Amazon datasets from distinct categories and first used binary classification to predict a "high" or "low" rating. In addition, the author attempted to find a more exact prediction using multi-class classification and logistic regression. Also, she trained and tested Naive Bayes, SVM, and Random Forest classifiers. Martin found SVM and Naive Bayes to be the most successful classifiers but noted that the binary classification also performed quite well for the other product category. Her conclusions were mixed due to differing results across product categories.

Finally, Park [3] analyzed aspects of product reviews across five categories and looked at their relevance to review helpfulness. The author then used four mining methods to find the best predictor for each product type. Park found that product differences mean algorithms need to be different across product categories. The author also concluded that the vector regression method was the most accurate predictor for each of the five categories.

IV. Conclusion

Considering both the EDA and the literature review, we have mush to ponder in developing our long-term goals of sentiment extraction for product review analysis. Indeed, it appears from the dataset, and the star_reviews that some of the review assessments may be artificially inflated, yet more study is necessary. Conclusions will be made after more work has been done in building and testing the various models. In addition, many of our data attributes are highly skewed, and this must be accounted for. At this stage we cannot make any hard conclusions on predictions, but report that we should have basic machine learning completed in the coming days.

References

[1] M. Martin. Predicting ratings of amazon reviews - techniques for imbalanced datasets. Master's thesis, Université de Liège, Liège, Belgique, 2017.

[2] G. Ocampo Diaz and V. Ng. Modeling and prediction of product review helpfulness: A survey. In *Proceedings of 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 698–708, 2018.

[3] Y.-J. Park. Predicting the Helpfulness of Online Customer Reviews across Different Product Types. *Sustainability*, 10(6):1–20, May 2018.