Designing and Building an Optimal Review Ranking System for Amazon

Mike Cai
mc6gb@virginia.edu

Yilin Huang
yh3mw@virginia.edu

Ivan Wang
zw4fc@virginia.edu

Haoxiang Zhang
hz4tt@virginia.edu

1 ABSTRACT

With the rapid growth and vast availability of the Internet to mass consumers, user generated content has become the one of the most popular way for businesses to engage with their prospective customers. Amazon realized the importance of UGC way back in the 90s, and it led the way with user-generated reviews to boost trust in their online products. On one hand, UGR is a great way of boosting consumers' trust in a product or service by democratizing the sources of information about it. By letting people like you and I to post reviews about a particular product we knew, bought, or used before, UGR can effectively crowd source the information about that product in a relatively accurate and unbiased fashion. Moreover, it can sufficiently eliminate the dichotomy between customers and brands by building communities that band current and potential users together.

However, the sheer volume of online content along with the increasing number of reviews related to just a single product can be overwhelming. There are many factors that influence the quality of a given review. For instance, there have been numerous scandals on the Amazon's Sellers Marketplace about how certain owners manipulated their reviews by outsourcing it to groups domestically and internationally, which specialize in writing fake reviews and giving false 5 star ratings. This just shows how user-generated reviews can be easily manipulated and are not trustworthy all the time. In contrast, it is just as likely for certain owners to incentivise the same group of writers to post fake bad reviews on a product of their competitors. Because of the reasons above, UGR can make things harder for individuals as well as business owners to locate the best reviews and understand the true underlying quality of a product.

In this research paper, we explore and examine the important attribute related to reviews from both a content-wise perspective as well as from a poster-credibility aspect in order to understand the perceived value they provide. We present the results and findings on the Amazon Review Data-set 2018, which contains all review data in Amazon from May, 1996 to October, 2018. The raw review data has 233.1 million reviews, which has a size of 34GB, but we only conduct analysis on the subsection for the purpose of this research. Our approach explores multiple aspects of review text, such as objectivity levels, time-decay degree, and more. In addition, we also include poster credibility metrics such as review consistency and legitimacy in our multiple reviewer-level features analysis. This paper dives into each of those aforementioned aspects in detail.

This paper aims to integrate information retrieval, econometric, text mining, and predictive modeling altogether toward a more comprehensive model to estimate the quality of user-generated reviews and thus provide a helpful ranking to them.

2 INTRODUCTION

2.1 Background

User-generated reviews are crucial to shoppers. As it turns out, every nine out of ten customers read online reviews before visiting a business and trust them as much as personal referrals and recommendations. According to the Nielsen Global Consumer Confidence Index, about 92% percent of consumers trust organic user-generated content more than they trust information gained via traditional advertising because of its authenticity.

2.2 Motivation

We believe that user-generated reviews are very important for consumers who are not familiar with certain products to make informed decisions. Unfortunately, a large number of reviews for a single product may also make it harder for individuals to evaluate the true underlying quality. Due to this sheer volume, especially for popular products and categories, finding the right ranking method to show the most relevant information to a potential customer could be tough to crack.

On the other hand, user-generated reviews could provide business with immense economic value. The seller of a particular product could definitely benefit from a well written, accurate, and informative review about the product that can convert the customer with matching need and generate sale volume.

2.3 Importance

More often than not, many online shopping sites only provide some simple techniques like rank by date posted or by helpfulness votes. Due to the sub-optimal methods that is prevalent in the current marketplace, we see significant value in researching and developing a ranking mechanism that could filter out the most relevant and helpful reviews a customer would want to know about a brand or a product, and we believe that it could build more trust and credibility with new customers. Meanwhile, finding a better ranking algorithm could also empower sellers and manufacturers of different products to identify the reviews that are more influential on customers' buying decisions, to examine the content of these reviews, and to potentially maximize conversion rate.

2.4 Research Question

To provide an effective and objective ranking algorithm for ranking product reviews, we have the questions below:

- What is a high quality review?
- What factors can help us determine the quality of a review?
- How does reviewers' credibility affects their reviews?
- What standards constitute a review as helpful and valuable?
- Why are those factors and standards important for a review to have?
- Is there well-defined metric to measure a particular ranking algorithm that would deemed to be effective?

In this paper, we will analyze a subset of review data from Amazon (from May, 1996 to October, 2018) to find a more comprehensive ranking algorithm for product reviews.

2.5 Data Set and Variables

Data set URL: https: //nijianmo.github.io/amazon/index.html

2.5.1 *Data Description.*

The dataset we chose is Amazon review dataset 2018 1 . It includes all review data in Amazon from May, 1996 to October, 2018. The raw review data has 233.1 million reviews, which has a size of 34GB. Considering the scale of our project, we only need a subset of the raw data. The website is friendly enough to provide some small pre-filtered dataset for experimentation. The dataset we consider using is called 5-core dataset. These data have been reduced to extract the 5-core, such that each of the remaining users and items have 5 reviews each, so users who rarely post reviews and items that is not popular enough to receive at least 5 reviews are filtered out.

2.5.2 *Data Format.*

The data format is one-review-per-line in json. It has 10 categories in total.

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- vote - helpful votes of the review
- style - a dictionary of the product metadata
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (UNIX time)
- reviewTime - time of the review (raw)

2.5.3 *Data Sample.*

The subset of the metadata we selected is the per-category data on musical instruments department because it only contains 231,392 reviews in it and this relatively compact size would fit the scale of our analysis the most.

Below is an except of the reviews:


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "vote": 5,
  "style": {
    "Format:": "Hardcover"
  },
  "reviewText": "I bought this for my
      husband who plays the piano.  He is
      having a wonderful time playing these
      old hymns.  The music  is at times
      hard to read because we think the book
       was published for singing from more
      than playing from.  Great purchase
      though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


2.6 Challenges

The first challenge we encounter is the selection of features for determining the quality of a review. After the features are selected, we will need to conduct feature reductions because too many features can lead to over fitting and cause confusion. Secondly, without using reviewers' private information, we will need to discover a method to quantify their credibility with the current data we have obtained. One of the hardest part of performing review ranking on this Amazon dataset is the lack of user information that is composited and comprehensive enough to query on. Since the dataset we select is structured but fragmented pieces of review attribute, we would need to provide and create query information in order to rank the returned review documents. Besides these initial challenges, we would then have to decide which kind of ranking algorithm would fit the best to the scenario, and make decision regarding to the trade offs among different methods.

3 RELATED WORK

One previous work on this topic is to design a novel review ranking system by focus on predicting usefulness and impacts. It was conducted by a team from NYU department of information science, and they proposed two ranking mechanisms for user-generated product reviews: a consumer-oriented mechanism that would rank the reviews according to their expected helpfulness, and a manufacturer-oriented mechanism that would rank them according to their expected effect on sales. Their research method was a combination of econometric analysis as well as text mining techniques, both of which focus on subjectivity analysis. The result of the paper showed that subjectivity analysis can give useful clues about the helpfulness of a review and about its impact on sales.[1].

Similar to the work of Ghose et al., we will combine two ranking mechanisms for ranking product reviews. Our purpose is to present high quality reviews for consumers so both of our mechanisms will be consumer oriented mechanism, ranking the reviews according to their helpfulness to consumers who are trying to make purchases decisions. When ranking a product review, not only will we take the quality of the review into consideration, but also the credibility of the reviewer. Taking such aspect into consideration allows us to better filter out fake reviews and counterfeiters that deceives

[1] <Ghose, Anindya, and Panagiotis G. Ipeirotis. "Designing novel review ranking systems: predicting the usefulness and impact of reviews." Proceedings of the ninth international conference on Electronic commerce. 2007.>

customers into buying products which are unlike with how they are described in the reviews.

Another related work we find is *Ranking online consumer reviews*, an excerpt from the journal *Electronic Commerce Research and Applications*. The purpose of this study is to rank the overwhelming number of reviews using a regression model that can predict their helpfulness scores. Features that are selected into the model are extracted from review text, product description, and customer Q-and-A data. The system categorizes reviews into high and low quality using a random-forest classifier, and it would then compute the helpfulness scores of the high-quality reviews using a gradient boosting regressor. The study excluded the low quality reviews out since they would have low predicted helpfulness scores and would not be included in the top k reviews list. Finally, the proposed system would rank the placement by predicted helpfulness score and place higher quality reviews on the top so that they would become more visible to the consumers. The experiment was ran on data from two popular Indian e-commerce websites and the findings indicate that inclusion of features from product description and customer Q-and-A data improves the prediction accuracy of the helpfulness score significantly [2].

In our study, the helpfulness score of a review is determined by the quality of the review and the credibility of the reviewer. When determining the quality of a review, the features we selected for predicting helpfulness score are review text length, image inclusion/exclusion (includes image or not), verification, time of the review, etc. Features in Saumya et al.'s work included information from customer Q-and-A data. However, information from customer Q-and-A date can be subjective. The helpfulness score of a review is then dependent on a subjective piece of information. This can cause subjective ranking of reviews. The features chosen in our studies are more objective and extracted from the reviews themselves, such as text length. Therefore, our rankings will be more reliable and less prone to bias.

4 PRELIMINARY WORK

Our preliminary work includes two sections. In the first section, we built an inverted index for all the reviews and reviewers, creating the bag of words. In the second section, we scraped a set of ranked reviews from Amazon to help us determine the parameters for our algorithm.

4.1 Building Inverted Index

4.1.1 *Data Cleaning.*

There were 10 total variables for each review documents in *JSON* format. Other than those variables that corresponds to string, there were some variables that need modifications before saving them to a review document and build the index.

- **Image** comes in as a list of URL links, and the majority of the reviews do not have any image, which would leave this field empty on the JSON file. We parsed the list and reassigned the corresponding number of images to the field.
- **Verified** is formatted as string and we convert it into dummy variable so that verified reviews would have a value of 1 and non-verified with 0.

4.1.2 *Inverted Index for Reviews.*

The first set of inverted index, *indices*, was built using the review data from a specific product. We filtered out review documents that do not match our targeted product number and only added review documents of the specific product to the corpus. The bag of words of a review document was made by tokenize *summary* and *reviewText* of the input JSON file. The index was built based on this corpus. All parameters of a review document were added to the index as fields, making them available to access after retrieval. Moreover, a new variable *content*, made by combining *summary* and *reviewText* of a review document, was added to the index as a field. *content* is a field that we were going to search in.

```
if(review.getString("asin").equals(asin)) {
    doc.setImage(review.get("image").length);
    doc.setOverall(review.get("overall"));
    //...
    doc.setBoW(tokenize((doc.getReviewText()
        +"␣"+doc.getSummary())));
    m_corpus.addDoc(doc);
}
```

4.1.3 *Inverted Index for Reviewers.*

The second set of inverted index, *useridx*, was built solely for storing user information. Unlike *indices* where we only kept review documents from a specific product, *useridx* contained every review document in this amazon dataset. In this case, the bag of words was set to *reviewerID* of a review document so that we could later use a reviewer ID as a query to retrieve information of a specific user. All parameters except *reviewerText* and *summary* of a review document were added to the index as fields. *reviewerText* and *summary* were left out for two reasons. The first reason is because adding *reviewerText* and *summary* will take a large disk space and a long index-building time since *reviewerText* and *summary* contains majority of the text of a review document. The second reason is because we were not going to need these two parameters due to the query generation method we chose, which will be discussed in the next section.

4.2 Parameter Extraction for Algorithm

To understand what separates a great product review from the rest, we decided to conduct an empirical analysis of the amazon review data by looking at one of the best sellers currently in the musical instrument department. (https://www.amazon.com/OneOdio-Adapter-Free-Headphones-Professional-Telescopic/dp/B01N6ZJH96/ref=cm_cr_arp_d_product_top?ie=UTF8)

4.2.1 *Product Selection.*

The product we selected is the #1 best seller DJ headphone, and it has more than 4000 shopper's reviews with an average rating of 4.7/5. Because of the popularity of this product, there is an abundance of its UGR contents for us to analyze the factors that contribute to the quality of a review.

[2] <Saumya, Sunil, et al. "Ranking online consumer reviews." Electronic Commerce Research and Applications 29 (2018): 78-89.>

4.2.2 *Web Scrapping.*

For most of the product review page, the shoppers are only interested in the top 50 reviews as it is unlikely for users to flip more than a couple of review pages before making up their minds. Therefore, we crawled and scraped the top 800 reviews for this product as it is approximately the top 20 percent of the total review rank and should be sufficient enough for us to understand both the contributing factors for top reviews as well as the pattern in general.

The data was crawled using *Scrapy*, which is an open source package and collaborative framework for extracting the data from websites. The original version of the data was stored in CSV format with the fields including author, review title, review content, helpful vote, rating, and whether or not it is verified.

The web-scraper-order column represent the time stamp at which a review is scraped. After the data is properly sorted and cleaned, the scraped review data results in the below format:

title	date	content	rating	verified	vote	rank
Super "bang-for-the buck"	Reviewed in the United States on June 4, 2018	This is an initial out-of-the box review:\nFir...	5.0 out of 5 stars	Verified Purchase	940 people found this helpful	1
Quality, High Fidelity, Comfortable, and a pri...	Reviewed in the United States on July 31, 2018	I bought these last year and let them sit on m...	5.0 out of 5 stars	Verified Purchase	315 people found this helpful	2
Clean, full sound with hearing compensation, a...	Reviewed in the United States on October 8, 2019	The OneOdio headphones have a very clean sound...	5.0 out of 5 stars	Verified Purchase	176 people found this helpful	3
Nice Product Awesome.	Reviewed in the United States on December 8, 2017	Beginning with the sound it is spectacular the...	5.0 out of 5 stars	Verified Purchase	137 people found this helpful	4
Great comfort and sound	Reviewed in the United States on July 20, 2019	I just got these and will use them only with m...	5.0 out of 5 stars	Verified Purchase	83 people found this helpful	5

Figure 1: Cleaned Amazon-Review-Data

4.2.3 *Feature Engineering.*

After investigating into Figure 1, it is easy to conclude that this raw form of data from the scrapped *JSON* file would require further data cleaning and feature extraction to provide any sort of analytical value. Thus, we conduct the steps below:

1 We first combine title, which is the summary of the review text, and content, which is the bulk portion of the review together as a single review feature in text format.

```
review = list()
for i in range(len(title)):
  review.append(title[i] + " " + content[i])
```

2 We then cleaned up the date format since the data only provides when a review was posted in string format, and we would like to know how many days have past since originally purchased so that we could apply time decay factor to each review and place more importance on qualitative reviews that are posted more recently. We strip out the sub-string that contain the term "$fullmonthname, dayandyear$" and convert it to the date time format:

```
time = list()
for i in range(len(dates)):
time.append(dates[i].replace('Reviewed in the
United States on ', ''))
new_date = list()
input_format = "%B %d, %Y"
output_format = "%Y, %m, %d"
for t in time:
nd = datetime.strptime(t, input_format).
strftime(output_format)
new_date.append(nd)
# converting string to datetime object
dt = [datetime.strptime(x, output_format)
    for x in new_date]
```

The original date value such as "*Reviewed in the United States on June 4, 2018*" would turn into the format of "2018, 06, 04" We then use the current day timestamp to subtract the previously found date-time to get how many days have passed since the posting, and the same methodology is applied in the final scoring function with $time_past$ passed in month as its unit.

3 We derived three review related features from the review text: $reviewlength$, $reviewsentiment$, and $reviewsubjectivity$, as we believe that those three features capture the amount of information, the neutrality degree, as well as the objectivity level of a review, and thus could provide significant value to our final scoring algorithm.

```
from textblob import TextBlob
review_polarity = list()
review_subjectivity = list()
for r in review:
  blob = TextBlob(r)
  sentiment = (blob.sentiment.polarity+1)/2
  subjectivity = blob.sentiment.subjectivity
  review_polarity.append(sentiment)
  review_subjectivity.append(subjectivity)
len(review_polarity), len(review_subjectivity)
```

4 The same string operation on *date* is applied on *vote* as well so that only the numerical part of the "x people found this helpful" statement would be left as value to reflect how helpful shoppers think of a review as.

After the data prepossessing and text mining, we derived a set of the new parameters for further regression and feature analysis and has a header shown in Figure 2:

- review length - the length of the review
- polarity - the tone of voice of the text review
- subjectivity - the level of how the judgement is formed based on personal opinions
- days past - UNIX review time
- rating - the product rating given by the reviewer
- image - number of images in the review
- helpful vote - votes from other reviewers/web-browsing users which indicates the helpfulness of the review
- verification - the review is verification or not

Rank	Review Text	Review Length	Review Polarity	Review Subjectivity	Days Past	Rating	Helpful Vote	Verified Purchase
1	Super "bang-for-the buck" This is an initial o...	354	0.6363265467171717	0.5221359427609428	688	5.0	940	1
2	Quality, High Fidelity, Comfortable, and a pri...	280	0.5693683862433863	0.4903968253968254	631	5.0	315	1
3	Clean, full sound with hearing compensation, a...	377	0.5975	0.4268115942028986	197	5.0	176	1

Figure 2: Processed Amazon-Review-Data

4.2.4 *Exploratory Analysis of Parameters.*

In order to create an optimal and objective ranking algorithm, we first need to understand the value each parameter attribute to the quality of a given review and then evaluate how Amazon ranks its reviews for a given product. Through uni variate analysis of the features from the processed data set, we are able to get a glimpse into both aspects, and several findings reveal that the existing ranking system adopted by Amazon can be improved as we will get into with our equation later in this paper.

- **The Review Length** feature is very important to a ranking model because it is the upper bound and proportional to the amount of information a review can capture. The distribution of the review length attribute is heavily right-skewed, as shown in Figure 3 below.

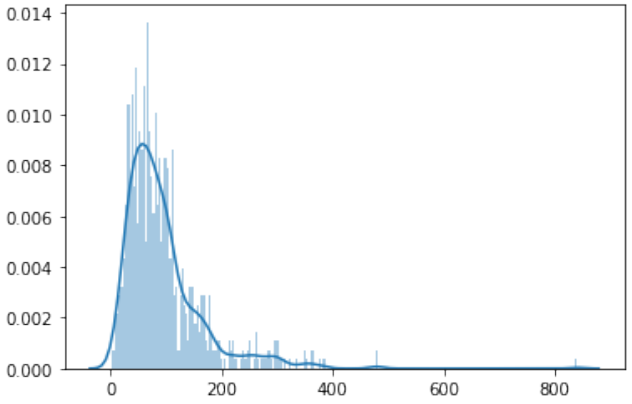

Figure 3: Histogram of Review Length

The review length attribute has an average value of 95.11 and a standard deviation of 72.36. Of the top 20 reviews for this product, 17 are above this average level, and of those, 15 are more than a standard deviation away to the upside. That lead to us believing that review length is the most important metrics amazon use when ranking its product reviews.

However, a longer review should not necessarily be the equivalent to more or better valuable content. Once a review reaches a certain length, the marginal amount of information gained would inevitably start to diminish because of repetitiveness or over granularity, both of which can make the experience of reading a review less enjoyable and less efficient.

Due the reason above, we attempt to de-skew the data by taking the natural log to make it more normally distributed, shown in Figure 4, so that reviews with overly lengthy text would not be favored disproportionately than those shorter ones. This will be passed into our final ranking score equation.

- **Review Sentiment:** The protocols adopted from *TextBlob* function categorize review sentiment into a bracket of two: *Polarity* and *Subjectivity*. The first metric focus mainly on the direction of an attitude toward a product in terms of positive, negative, or neutral feelings detected in a review;

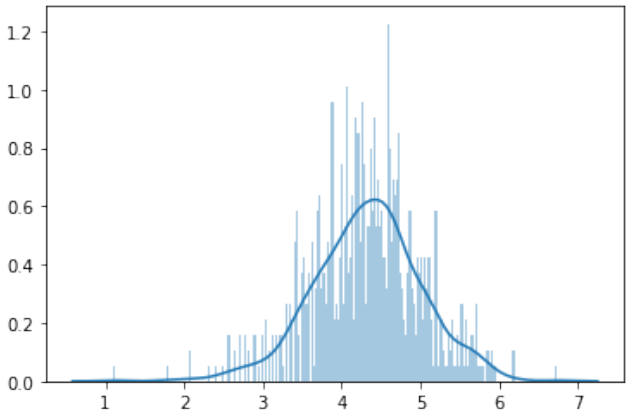

Figure 4: Histogram of Normalized Review Length(Log)

the later refers to whether a review carries a more subjective or objective tone.

  - **Polarity** by default comes in with value ranges from −1 (most negative) to 1 (most positive), with 0 being neutral. Our preliminary analysis showed that most of the reviews are distributed on the positive sides, matching the expectation of the high star rating average we discovered earlier because of the popularity of this product.
  - **Subjectivity** ranges from 0 (most objective) to 1 (most subjective), and the distribution for this product is also slightly more subjective, with an average of 0.5616 and a standard deviation of 0.1154.

We re-scaled this *polarity* index by shifted its value up 1 unit and then divided it by 2 so that the new index would range from 0 (most negative) to 1 (most positive), matching that of the *subjectivity* index. The joint distribution of them is shown in Figure 5 below:

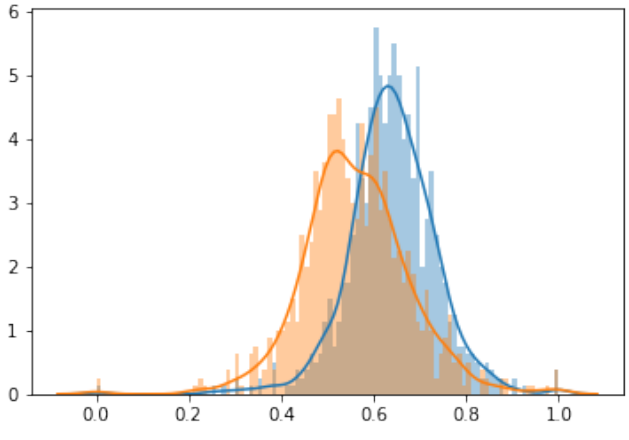

Figure 5: Joint Distribution of Sentiment

**The orange histogram representing the subjectivity and the blue one representing the polarity level*

From the graph above, we can see that the general sentiment for this product is positive (with an average polarity of 0.63 and a standard deviation of 0.09) and relatively objective with Subjectivity around 0.56.

- **Duration of Review** Some of the top ranked reviews were posted a year or two ago. Outdated reviews cannot represent or speak for the current status of the product. A factor that caused the outdated reviews to have such high ranking is their high number of helpful *votes*. The current algorithm that heavily weighs on the number of *votes* when ranking will bury high quality reviews that were posted recently.
- **Helpfulness Vote** variable is extremely unevenly distributed with the top 50 votes accounting for 77.108 percent of the total helpful vote counts. This indicates that Amazon place a disproportionate emphasis on the number of helpful votes in its ranking algorithm, but we think that it is an overkill. When number of helpfulness vote over occupies other intrinsic value of a review text, new reviews that do not yet have a high number of votes will get ranked unfairly against past reviews that has, so we would reduce the importance of this metric in our model by reducing weight as we will get into later with our findings from the sample Amazon reviews.
- **Verified Purchases** has an important effect on the product review ranking. Of the top 800 reviews we extracted from the total of more than 4000, almost all of them are verified and a few are listed as "VINE VOICE", which are the most trusted reviewers Amazon invites to post opinions about newly released items to help their fellow customers make informed purchase decisions. However, given the rare occurrences of those type of vine review, we will only consider the verified review from the unverified, and this factor will play an important role in our equation as we will get into later on in this paper.

4.2.5 *Amazon's Way of Ranking.*

By looking into the top 50 reviews for this product, we discovered some potential patterns and trends of the Amazon ranking system. As discussed above, the there is enough evidence to believe that Amazon place a heave emphasis on longer review than shorter ones, but 10 out of the top 50 reviews have abnormally short review length in comparison to their neighbors.

A closer investigation into those specific reviews reveal that they all have significantly higher sentiment score from the average in terms of review polarity. We dive into a most representative example below and analyze the implication of it to our own algorithm:

- **Sample Review:** *"Nice Product Awesome. Beginning with the sound it is spectacular the levels of low and treble that allows you to use it is spectacular, its construction is of very good quality, it comes with two cables of very good aspect and quality. It is a product that gives a very good benefit for its coast"*
- **Analysis:** This review is ranked 4th overall. Nonetheless, this piece of text only contains 55 words in length and is obviously lacking in contextual and descriptive information about the quality of the headphones. What is also obvious about this review is the extensive use of positive nouns like quality and benefits as well as the overwhelming presence of superlatives adjectives such as nice, good, awesome, and even spectacular while not mentioning any downside to purchasing this product. Those words give this review text a polarity score of more than 0.82, which is 2 standard deviation higher from the mean neutrality of the reviews for this product in general. Besides the lack of neutrality in this review, it also contains some blatant grammatical mistakes, such as the misspelling of the word "cost" as well as information redundancy.
- **Implication:** The review excerpt above fails to address both the pros and cons of the product in comparison to some other top ranked reviews yet is ranked higher than many other quality reviews that give more informative and less exclamatory tone. Because this phenomenon is quite prevalent among the top 50 reviews, we hypothesize that this mechanism is intentionally designed to drive more product sale and would benefit the sellers more than the shoppers.
- **Correlation Analysis:** To validate our hypothesis, we took the log of the inverse review rank of the reviews to get a ranking score for each review, and took the z-score of it and plotted against the normalized review sentiment, the result proved our hypothesis about how Amazon go about ranking their product. From Figure 6, there is a clear positive association between the sentiment of a review and its ranking score, meaning that when all else being equal, the higher a sentiment is, the more likely a review will be ranked up to boost its online sale.

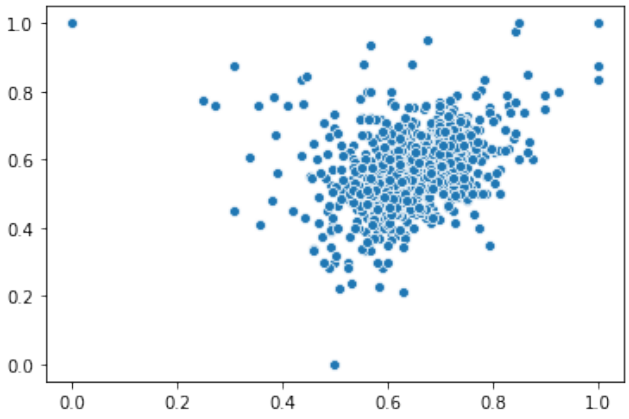

Figure 6: Scatter Plot of Review Sentiment vs. Ranking Score

In order to utilize review sentiment to give shoppers a more unbiased review ranking, we decide to calibrate the review sentiment index by taking the inverse of the absolute distance a given review sentiment score (*polarity*) from the mean review sentiment of that product query. By doing that, our final scoring model will effectively prevent the effect of those overly positive review from altering our final ranking.

Besides, our bi-variate analysis showed no correlation between the subjectivity level of a review with its quality, so we will construct sentiment score described previously using the *polarity* index only.

5 ALGORITHM

We take a three steps process in constructing our final scoring model by first internalize the findings from our review analysis, by then assigning default weight for each parameter, and finally by putting them back together for our algorithm.

5.1 Intuition

Based on the research findings from the scrapped amazon review data, we think that the quality of a particular review has three factors affecting it, namely they are intrinsic value, extrinsic value, and time-related value. Moreover, those three main factors can be further divided into the following components:

- **Intrinsic Value:** can be directly derived from a piece of review text itself, and we have determined those factors to be review length (Rl), review sentiment score index (St), helpfulness vote (Hv), and the relevance of a review to the user query (Rq).
- **Extrinsic Value:** this indirectly affect the quality of a review base on factors outside of the review text or query matching. Our model will include User Credibility Metrics (Crd), whether a review is verified or not (Vrf), and the number of images that come along with a review (Img) as the three extrinsic factors.
- **Time Decay Value:** we represent this factor as the Duration of Review (DoR) to account for the legitimacy and quality of a review by discounting it to present time (unit in months).

5.2 Bench-marking Hyper-parameter

Before we put the scoring equation together and assign specific weights to different hyper-parameter, we designed a machine learning pipeline to train the scrapped amazon review data in order to gain a better understanding of how it allocates parameter weights.

5.2.1 *Multivariate Regression.*

The regression has 7 independent variables (x_1: Review Length (Rl), x_2: Polarity, x_3: Subjectivity, x_4: Duration of Review (DoR), x_5: rating, x_6 Helpful Vote (Hv), x_7 Verified (Vrf)), and the target variable is the modified ranking score as described in the previous section. We get the following OLS results:

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0066	0.000	13.822	0.000	0.006	0.008
x2	0.7198	0.400	1.798	0.072	-0.066	1.505
x3	-0.1657	0.289	-0.573	0.567	-0.733	0.402
x4	0.0001	0.000	0.935	0.350	-0.000	0.000
x5	-0.0558	0.029	-1.894	0.059	-0.114	0.002
x6	-0.0004	0.001	-0.447	0.655	-0.002	0.001
x7	4.8589	0.243	19.965	0.000	4.381	5.337

Figure 7: OLS Results

From the regression coefficient, we can see that Amazon's way of ranking places a very high importance on x_1: Review Length (Rl) and x_7: Verified (Vrf)), as there p-value are below 0.05, x_2: Polarity and x_5: rating as those coefficients have $p-values \leq 0.1$, indicating the statistical significance of them.

The table shows that when polarity is present in the model, the effectiveness of subjectivity is greatly reduced either due to no correlation or due to multicollinearity, which would justify our decision to leave subjectivity out of our model.

We will base our model off of those coefficient but adjust accordingly. For example, ratings will not play a role in our model since higher rating should not mean better product review quality whereas duration of review would have a bigger weight as a multiplicative factor in our final equation.

5.2.2 *Random Forest.*

The random forest model builds and ranks all features importance in a classifier and gives us valuable hint as of how much weight we should assign to each parameters as a benchmark, as shown in figure 8 below:

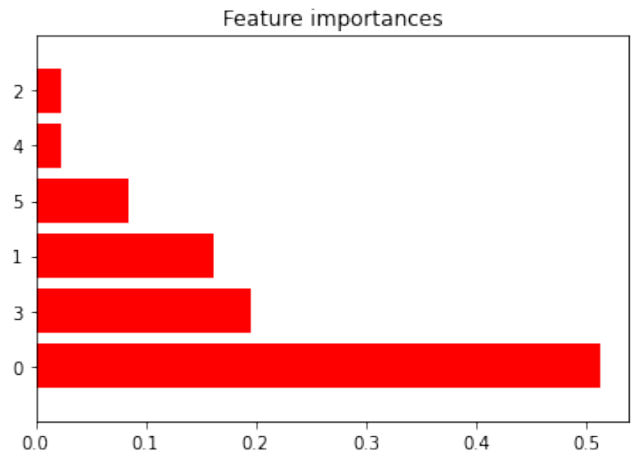

Figure 8: Feature Importance

5.3 Scoring Model

$$Score = DoR \cdot Vrf \cdot (Crd \cdot (w_1 \cdot Rl + w_2 \cdot St + w_3 \cdot Vt + w_4 \cdot Rel) + Img)$$

$w_i \in \mathbb{R}$ represents a customized weight.

5.4 Definition of parameters

Using the default parameters we have for each document, we generated new parameters to use in our scoring function.

Definition 5.1. *DurationOfReview*, DoR is defined as follows

$$DoR = \frac{1}{\log_2(MonthsElapsed + 2)}$$

This is the time decay value mentioned previously, where $Months-Elapsed$ is the number of months between current time and the time of posting the review. More recent reviews would have a higher value than older reviews. Using $MonthsElapsed$+2 prevents more recent reviews having significantly higher scores.

Definition 5.2. $reviewerAverageVote$ is the average number of helpfulness votes an user receives for each of his/her review.

The credibility of an user, Crd, is defined by:

$$Crd = 1 + \log(1 + \log(1 + \text{reviewerAverageVote}))$$

The purpose of the credibility factor is to determine whether this user is an "expert" in writing review. If the average votes an user receives in each of his/her review is high, then it means that they usually post helpful reviews, thus making him/her have high credibility. Using double log allows us to smooth out the numbers and prevent this factor overwhelm the rest of the formula. Users with no previous reviews will have a 1 in this term.

Definition 5.3. $Sentimentscore, St$, is defined by:

$$St = \frac{2}{1 + |sentiment - AverageSentiment|}$$

where sentiment score is calculated using Stanford CoreNLP, and returned as a number between 0 and 4, measuring the positiveness or negativeness of review. The *Average Sentiment* represents the mean sentiment for all reviews of this product. Having a sentiment score close to the average means this review has a fair assessment of this product and thus will receive higher score in this term.

Definition 5.4. $Relevancyscore, Rel$, measures the relevancy of a review to the query. We used the following retrieval function, inspired by the paper [3]:

$$Rel = \sum_{t \in Q \cap D} C_t^Q \cdot \frac{C_t^D}{C_t^D + 0.5 + \frac{0.5 \cdot |D|}{avdl}} \cdot \log \frac{N+1}{df(t)}$$

The terms follow the naming convention in the paper, where t stands for the term, Q stands for query, and D stands for the review document, with summary and review text concatenated together. [3].

Definition 5.5. Other parameters, Rl, St, Vt, Img, and Vrf, used in the equation is defined by:

$$Rl = \frac{\log(reviewLength+1)}{\log(AverageReviewLength+2)}$$

$$Vt = \frac{Vote}{AverageVote}$$

$$Img = \log(imageNumber + 1)$$

$$Vrf = \begin{cases} 1 & \text{, verified purchase} \\ 0.1 & \text{, not verified purchase} \end{cases}$$

6 EXPERIMENT

6.1 Query for Retrieving Data

Although document ranking is not exactly the same as searching, where the user would input a search query every time, document ranking is always related to a query as well. To be more precise, this query should be representative of what a user want to see in the review. In real life, this query should be generated from some sensitive user information. Since it is impossible for us to access user information such as recent purchase, recent search, and recent viewed product, we have two methods that could generate a query without having access to any sensitive user information.

The first method is to generate a query by analyzing other reviews that a user posted. Similar to document analyzing stage, we could set a bag of words by tokenize all the reviews of a user. From this bag of words, we could use some of the most frequent and meaningful words to create a query string. This query string could be representative of what this specific user is looking for in other review documents. There are downsides of this method. Although it is guaranteed that we are able to find other review documents of a user in this dataset, there are many users who do not post reviews regularly even though they could be very active on Amazon. Therefore, in real life situation, it is not guaranteed that we could generate query based on this method and the information contained in this query is probably much less than query generated using sensitive information. Since our work is only for demonstration purpose, we decide not to dig deeper into generating personalized queries using such limited user information we have.

The method we used is to generate query by simulating users. We assumed the user would like to describe his/her information need using a keyword query (long battery life and good customer service). Although it is simple, this method worked well in this case because it was a good representative of what a user might be looking for. They could be very close to the query generated by the sensitive information in real life.

6.2 Results and Analysis

In order to validate our ranking algorithm, we ranked reviews for many products, each with a few different queries, including a default ranking where none of the documents has a match with the query. Below is a sample ranking result for query *sturdy product* on *Yamaha FC3A foot pedal*:

(1) Title: Took a chance, grateful I did.
Review Text: ...this pedal is very *sturdy* and does it's job exceptionally well, without any fluff to inflate the price. It's low cost, it's quality construction, and it does exactly what it says it does...a quality product at a very reasonable price.
Vrf: 1, St: 0.18685, Rel: 0.32344, Crd: 1.98356, Rl: 0.52802, Vt: 0.06614, Score: 1.29585

(2) Title: Great Stereo Volume/Expression pedal!
Review Text: ...it works flawlessly...as both a volume and expression...I use it mainly as a permanent fixture on my custom made digital only effects board for guitar...are still the same great quality...
Vrf: 1, St: 0.27959, Rel: 0.0, Crd: 1.88434, Rl: 0.49381, Vt: 0.26454, Score: 1.25040

(3) Title: Works Great with my Yamaha Reface
Review Text: ...this is a great pedal...a *sturdy* large pedal, even wider than my guitar wah pedal. I bought this for use with my Yamaha Reface...
Vrf: 1, St: 0.26214, Rel: 0.27157, Crd: 1.63087, Rl: 0.44160, Vt: 0.06614, Score: 1.16369

(4) Title: Great, takes some getting used to.
Review Text: ...works as advertised...able to use half-pedaling

[3] <Fang, H., Zhai, C. (2005). An exploration of axiomatic approaches to information retrieval. In SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 480-487).>

...You have a bit of travel on the pedal before it actually registers... and the half-pedaling effect seems to go to a full pedal early...If there was a way to adjust the sensitivity on this pedal...well worth having one of these...
Vrf: 1, St: 0.25961, Rel: 0.0, Crd: 1.82459, Rl: 0.50147, Vt: 0.16534, Score: 1.16034

(5) Title: No right angle plug or removable cord
Review Text: ...very good quality...it has an attached cord instead of allowing you to use your own 1/4" cord...need an right angle connection at the end which this does not have...Would not recommend if...
Vrf: 1, St: 0.17089, Rel: 0.0, Crd: 2.41319, Rl: 0.44682, Vt: 0.13227, Score: 1.12881

Since there is not a standard rank for us to base on, instead we made a few speculations manually and see if the results are reasonable decisions based on the information need and what is presented. For this specific query and product, the first and third ranking review was ranked 10 and 12 in the default ranking. All reviews default-ranked above 20 offer a similar degree of detail on different aspects of the product, so this is a reasonable ranking that prioritizes the information need while also providing holistic reviews on other aspects of the product.

It is also worth noting that there are three other reviews with matched terms for this query, and their ranks rose from 62 to 33, 74 to 42, 76 to 44 respectively. This shows that even if a review has matched terms, its ranking is still dependent on its other inherent features. From multiple ranking results, we found that the algorithm tends to rank a document about half its default rank when it has a matched term.

7 LIMITATIONS AND FUTURE WORK

As we previously mentioned, one major limitation is that we have no access to Amazon user data. This limitation could affect us in multiple ways. If we have the access to user data, we could generate personalized queries for document retrieving, rather than using queries we manually put in. This would make the document retrieving process more realistic. Moreover, we could compute a more comprehensive credibility score for each user. For now, the credibility score of a user is simply just the average number of votes per review the user received in the past. There are much more factors that could be taken in account other than votes: for example, number of items purchased in total and time since account creation.

The second limitation is that after creating the ranking model, we are not able to validate our model efficiently since there is not a hard standard for ranking product reviews. Optimally we would have human volunteers rank a large amount of reviews based on different information needs, and evaluate our ranking algorithm against the proposed ranking. This was not possible due to availability issues. As a result, we cannot tune the parameter coefficients because we cannot really determine if one ranking is better than the other. Currently, the way we determine if our ranking is reasonable or not is by checking the ranking results of our algorithm manually. In future studies, researchers might have to come up with a way to evaluate rankings.

Another future study direction is to look more into the review text of each review instead of features of it. The features are merely heuristics that could indicate a good review, but if we are able to analyze the text directly, we could have a more direct and therefore accurate evaluation of the reviews. One possible solution is to analyze the reviews sentence by sentence, and extract keywords to see the range of the review.

8 CONCLUSION

In this paper, we present a multi-feature based review ranking algorithm. We introduced many new factors for evaluation, including author credibility, recentness, polarity, relevance. Each of these new features are designed with a purpose of ensuring the quality of a review while still staying at a heuristic level of understanding. The algorithm also analyzes other helpful features of a review, including length, votes received, etc. All of these is critical in evaluating reviews, and weight on each feature determines it importance. The random forest gives us a good idea on how we should weight each of these features. The result is a novel review ranking system designed to provide an enhanced user experience with the review system.

REFERENCES

[1] Saumya, Sunil, et al. "Ranking online consumer reviews." Electronic Commerce Research and Applications 29 (2018): 78-89.
[2] Ghose, Anindya, and Panagiotis G. Ipeirotis. "Designing novel review ranking systems: predicting the usefulness and impact of reviews." Proceedings of the ninth international conference on Electronic commerce. 2007.
[3] Fang, H., Zhai, C. (2005). An exploration of axiomatic approaches to information retrieval. In SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 480-487).