Helpfulness of Amazon Reviews

Amazon is one of the largest ecommerce marketplaces in the world with millions of users making transactions daily. These users can review the products they buy which influence other users. The reviewer gets points in the form of votes depending on the helpfulness of the review. With enough points that reviewer becomes verified and qualifies for free items to review. The items depend on the type of products they were reviewing. I was curious about what makes a review helpful and whether there are factors that distinguish a review as helpful or not. Knowing this could lead to becoming a top reviewer which results in free products.

Data Wrangling:

The data comes from UCSD with reviews up to 2018. The data includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features). The data was divided into 29 categories - however I was unable to include all of the categories due to my lack of memory. I was limited to datasets below ~10 Million reviews, leaving me with 21 categories to process.

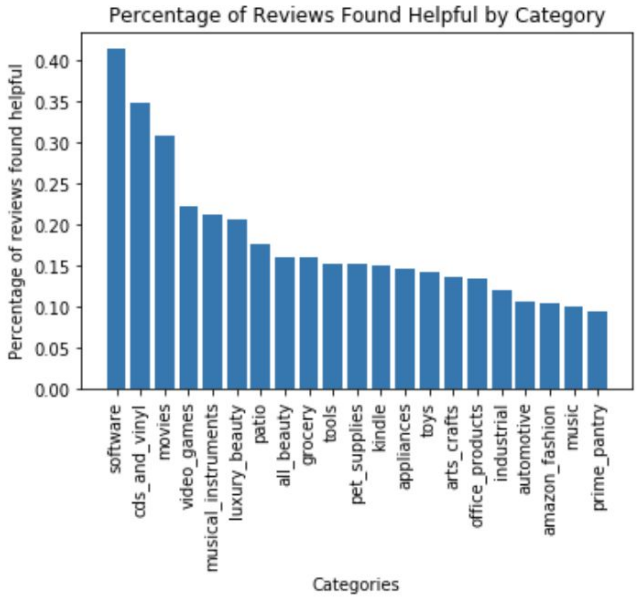
The goal was to take equal random samples from each category and combine them into one large dataset. For each category a sample of 3000 random products were selected through the ASIN (Product ID) that have at least 5 reviews. From these products only 5 reviews were selected randomly- resulting in a dataframe with 15000 rows. I then added a column indicating which category the product came from. These were merged with the metadata and saved as a csv file - repeating until there was a dataframe for each category.

All csv files had only relevant columns selected. I concatenated all of the dataframes into one final dataframe. After concatenating the datasets into a final dataframe, I created columns for lemmatized reviews and lemmatized summaries. I tokenized each review and lemmatized each word with its part of speech to reduce it down to its root form. I then combined the tokenized words back into its original form.

Exploratory Data Analysis:

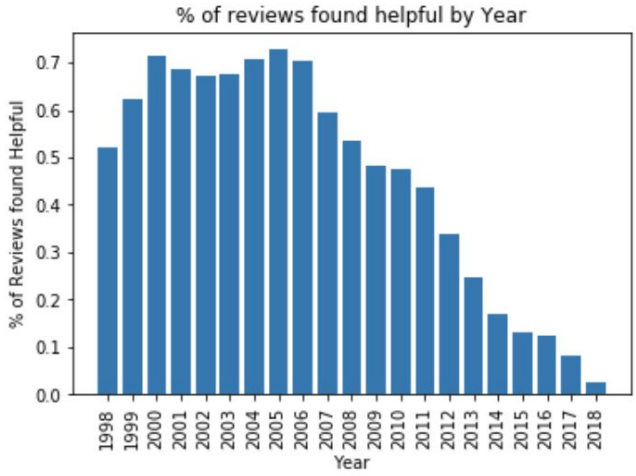
After cleaning the data I added a column to indicate whether or not a review was found helpful. This was done by simply seeing if it had any helpful votes and creating a separate binary column. I took the sum of helpful reviews and found that the percentage of reviews that are found helpful are about 17% of all reviews.

I split the data and found how many reviews are helpful by category:



We see that each category is different in their share of helpful reviews. The categories with the highest percentage of helpful reviews are Software (41.4%), CD's and Vinyl (34.7%) and Movies (30.7%) followed by Video Games (22.1%). The categories with the lowest percentage of helpful reviews are Prime Pantry (9.46%), Digital Music (10%) and Amazon Fashion (10.3%). There is a clear distinction between the categories for the amount of helpful reviews.

I also plotted time against the percentage of reviews found helpful:

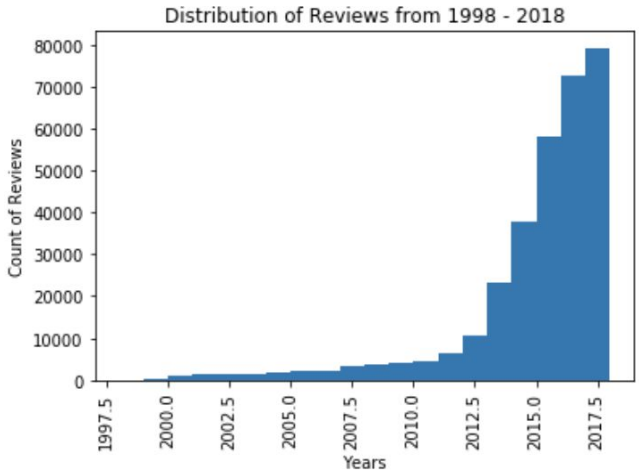


Year of posting appears to have a dramatic effect on the % of reviews that are found helpful. This appears to indicate that the length of time that a review is available on Amazon is a large factor in determining whether or not it gets helpful votes, with effects plateauing out after

around 12 years. To account for the bias introduced by the date posting, I cut out the data prior to 2013.

We see that older reviews are found more helpful than newer reviews. This is likely because it gives more consumers exposure to the review. There appears to be a decline in reviews found helpful after 2006.

I plotted the distribution of reviews based on the date posted:



The distribution of reviews are skewed to the left, with a majority of reviews from the dataset posted recently. I queried the data for reviews posted after 2013 and found that this contains roughly 79% of the data.

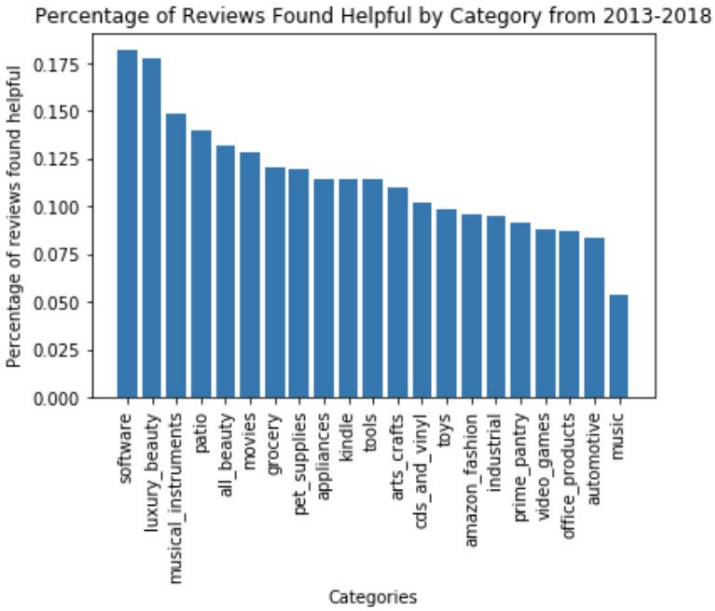
I looked at the quantity of reviews for each category:

prime_pantry	14874
amazon_fashion	14299
industrial	13686
arts_crafts	13576
automotive	13394
appliances	13338
all_beauty	13273
patio	13168
luxury_beauty	13058
tools	12940
pet_supplies	12727
office_products	12517
grocery	12417
toys	12189
musical_instruments	11771
kindle	11669
music	10354
video_games	8485
movies	8189
software	6278
cds_and_vinyl	5689

The four categories with the lowest amount of reviews are video games, movies, software and cds. Interestingly these categories had the highest share of helpful reviews before the older dates were filtered out. This underlines the importance that time has on review helpfulness. These categories may not have had the highest share of helpful reviews because they were digital, but because they had the highest amount of older reviews. I removed the older years

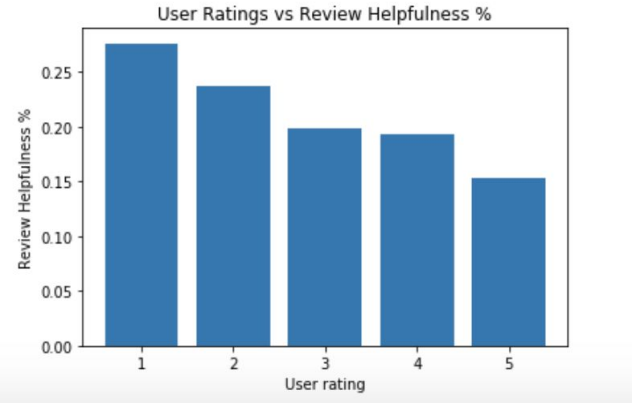
(2012 and prior) to observe what effect it would have on the percentage of reviews found helpful.

I plotted the share of helpful reviews by category after 2013:



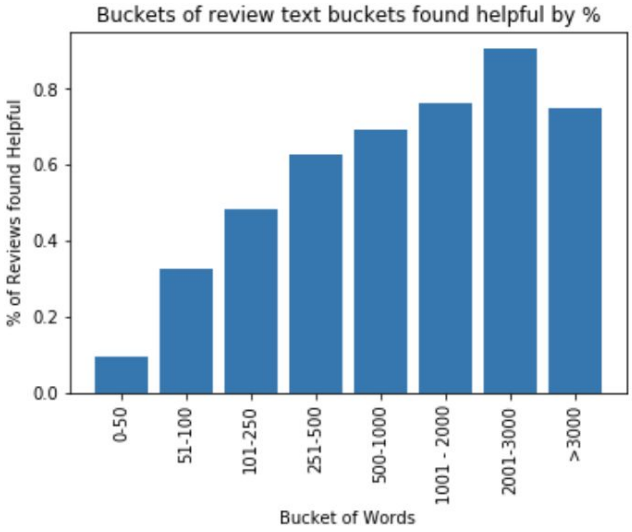
Without the older reviews - software is still on top with the highest percentage of helpful reviews however the next categories are luxury beauty, musical instruments and patio. The other 3 previous most helpful categories (movies, cds and video games) have all fallen out of the top 4, further highlighting the importance that time has on helpful reviews. We also see that the highest percentage of helpful reviews decreased from 41% to 18%.

After analyzing the effect that time has on review helpfulness, I plotted the percentage of reviews that are found helpful by the rating that each reviewer had given.



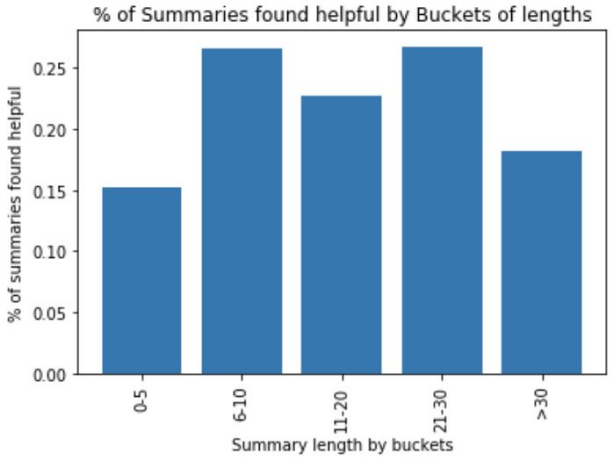
From the chart we see that a higher percentage of reviews that rate the product lower are found helpful. Reviews that are more critical of the product are helpful to consumers proportionately more than reviews that just praise the product.

I plotted buckets of review lengths to see if longer reviews resulted in a higher share in helpful reviews:



We see that longer reviews are typically found more helpful. A majority of reviews over 250 words are found helpful (62% and above). However, this peaks at between 2001-3000 word reviews, with helpfulness declining modestly after that.

I plotted buckets of summary lengths to see if there was a similar effect:



Unlike for review text, it appears that a longer summary length is not exactly indicative of a more helpful review. There appears to be a sweet spot between 6 and 30 words, with very long and very short summary lengths having lower helpfulness ratings.

Statistical Inference:

I conducted a chi-squared test to see if the frequencies of helpful and unhelpful reviews are different. For this I took a random sample of 10,000 helpful and 10,000 unhelpful reviews

and used the review length buckets for comparison. When conducted on review lengths, the chi-squared test yielded a p-value of 0.0 which makes it clear that helpful reviews are longer than unhelpful ones.

I also conducted a chi-squared test for the frequencies of helpful and unhelpful summaries. I used the same sample of 10,000 helpful and 10,000 unhelpful summaries and used the summary length buckets for comparison. When conducted on summary lengths, the chi-squared test yielded a p-value of 6.610e-98. Helpful summaries are significantly longer than unhelpful summaries.

Helpful/Unhelpful Words:

I used a Naive Bayes classifier to predict helpful reviews and helpful summaries then found words that had both the highest and lowest probabilities of being in a helpful review. The results varied for each category, review text and summary text.

I listed the words with the highest probability of being helpful with their probabilities and the words with the lowest probability of being helpful with their probabilities.

Helpful Probability by Word: Review Text

Helpful words	P(Helpful | word)
scherzo	0.89
sonata	0.88
seventh	0.88
brahms	0.87
bartok	0.86
richter	0.86
quartets	0.84
frontpage	0.83
schubert	0.83
poser	0.81
Unhelpful words	**P(Helpful | word)**
manifold	0.02
historia	0.02
drumstick	0.02
carburetor	0.02
goodie	0.02
reeds	0.02
libro	0.02
peavey	0.02
producto	0.01
sheath	0.01

We see that words that are listed in languages other than english (libro, producto) have low probabilities of being considered helpful. The helpful words are typically terms or names that are associated with the specific product (scherzo and sonata are latin words used in classical music). There are an unproportional amount of musical terms and brands in this list. I've then listed the helpful and unhelpful words associated with helpfulness for the summary text.

Helpful Probability by Word: Review Summary

```
Helpful words        P(Helpful | word)
            acquaint 0.85
            baritone 0.85
          challenged 0.85
                  96 0.80
           collision 0.71
                crop 0.67
              coffee 0.66
              crunch 0.66
              dances 0.65
               crazy 0.64
Unhelpful words      P(Helpful | word)
             confirm 0.04
              cooked 0.04
          categorize 0.04
           abilities 0.04
            creation 0.04
               cakes 0.04
             bedtime 0.03
              ballad 0.03
         compression 0.03
                babe 0.03
```

The summary text words overall are very different from the review text. I don't really see a pattern in the summary words.

Next, I performed the same analysis but with a breakdown by category:

Helpful Reviews: Music

	Helpful_words	Helpful_prob	Unhelpful_words	Unhelpful_prob	category
0	op	0.938433	chant	0.018605	music
1	adagio	0.833471	whenever	0.018605	music
2	di	0.812117	sea	0.018605	music
3	ii	0.803728	hearing	0.017530	music
4	walter	0.803728	blessing	0.017199	music
5	andante	0.784481	remix	0.015990	music
6	verdi	0.761050	cassette	0.014019	music
7	concerto	0.761050	lol	0.013206	music
8	cobham	0.761050	oldie	0.013206	music
9	liszt	0.747313	relate	0.012481	music

I looped through the dataframe to find the most predictive words per category for review text and found that the exact words are very different across each category, however the type of words across categories have similarities. Words that describe a family member are common for bad words across categories (nephew, grandson, granddaughter). It appears that reviews for products that were purchased as a gift for someone are rarely viewed as helpful. For good words I noticed that the words that are helpful are those that pertain to the specifics of each category. These words are usually specific brands or terms for each category.

Next I looped through the summaries for all categories:

Helpful Summaries: Appliances

	Helpful_words	Helpful_prob	Unhelpful_words	Unhelpful_prob	category
0	stainless	0.751827	fits	0.036487	appliances
1	avoid	0.751827	fit	0.033618	appliances
2	food	0.707907	arrive	0.032564	appliances
3	reliability	0.707907	replace	0.031712	appliances
4	beautiful	0.707907	original	0.029404	appliances
5	blue	0.707907	item	0.026802	appliances
6	stop	0.707907	four	0.024482	appliances
7	steel	0.707907	three	0.020125	appliances
8	cool	0.679551	stars	0.016704	appliances
9	garbage	0.645097	five	0.016694	appliances

For summaries I found in every category that the most common unhelpful words are numbers and stars (i.e. “Five Stars”).

After querying by dates later than 2013, I found the helpful and unhelpful words for each category:

Helpful words after 2013: Software

	Helpful_words	Helpful_prob	Unhelpful_words	Unhelpful_prob	category
0	aftershot	0.513592	easy	0.001616	software
1	32	0.363130	works	0.001528	software
2	plugin	0.232875	hard	0.001335	software
3	color	0.197672	great	0.001271	software
4	black	0.191968	love	0.001226	software
5	transcription	0.174357	found	0.001226	software
6	spot	0.174357	dvd	0.001146	software
7	written	0.174357	computer	0.000858	software
8	proficient	0.174357	game	0.000837	software
9	lightroom	0.174357	learn	0.000621	software

While the predictive power of helpful words are not very strong, the most interesting pattern emerged after removing the reviews prior to 2013; words that are subjective to the user are classified as unhelpful. Words like “great”, “love” and “perfect” are all unhelpful words while the helpful words are objective. They’re primarily words that are used to describe the product, not the user’s sentiment. This pattern also emerged in the summaries for reviews posted in 2012 and later.

Applying Machine Learning:

Predicting the helpfulness of an Amazon review is indicated by the column created earlier. I used different text classification models to predict the helpfulness of reviews. I selected a handful of models to choose from: Naive Bayes, Logistic Regression, Random Forest and Stochastic Gradient Descent Classifier (SGDC). I tuned the hyperparameters for all models using Grid Search Cross Validation before making predictions with the test set.

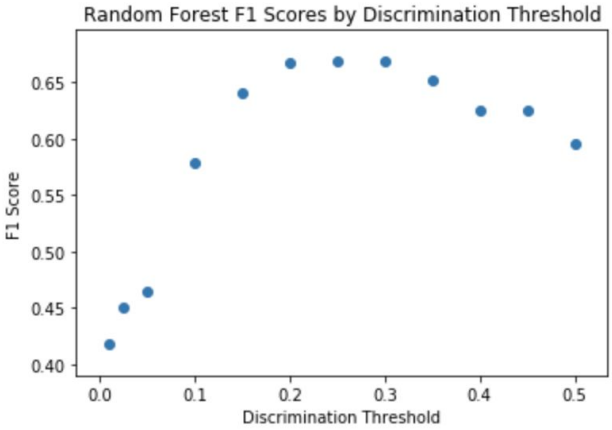
I split the dataset into a training set and a test set then identified the index for each. This way I could use the same training set and test set for models with different input variables. The first set of models I created used only the reviews, the second set of models used only the summaries and the last set of models used reviews, summaries and categorical features (review length buckets, summary length buckets and year). I measured the results of each model with an f1 score and logged the results.

In order to optimize each model's f1 score, I looped through a list of discrimination thresholds and compared each models' predicted probability against this. By manually setting the discrimination threshold I was able to optimize each models' performance by selecting a threshold that would yield the highest f1 score. I chose the threshold that resulted in each models' highest performance and logged the results.

I created a table of the results for the models:

	Review_Scores	Summary_Scores	Total_scores
NB	0.649979	0.612599	0.451159
RF	0.668417	0.590886	0.651031
SGDC	0.531218	0.563519	0.531218
Log_Reg	0.523929	0.565899	0.523929

We can see from the results that a Random Forest model that only uses reviews performs best. I've also plotted the f1 scores by discrimination threshold for the best performing model:



Since the Random Forest model with only reviews performed the best I used it again but segmented by categories to see if it would perform better if it were trained on specific categories rather than the overall dataset. I segmented the original dataset by category then queried those specific categories to see if their rows were in the training and test sets. This way I would be able to maintain the same training and test sets to compare the performance against the highest performing model. I used an f1 score to measure the performance of the category specific models against the generalized model used earlier for predicting on each separate category.

I created a table of the results for category specific models vs generalized model:

	category	category_specific	generic
0	office_products	0.599596	0.617719
1	toys	0.591861	0.639055
2	amazon_fashion	0.527179	0.584936
3	video_games	0.685761	0.692331
4	pet_supplies	0.597508	0.626997
5	all_beauty	0.584817	0.601474
6	automotive	0.590505	0.612326
7	tools	0.607845	0.626624
8	movies	0.712731	0.693148
9	music	0.661414	0.666858
10	grocery	0.590846	0.603313
11	kindle	0.516871	0.563855
12	software	0.677868	0.688860
13	musical_instruments	0.660593	0.661102
14	luxury_beauty	0.605751	0.598231
15	appliances	0.712780	0.693046
16	arts_crafts	0.703600	0.708822
17	industrial	0.603840	0.628368
18	prime_pantry	0.569972	0.593354
19	patio	0.639369	0.628803
20	cds_and_vinyl	0.725741	0.732843

The generalized model outperformed the category specific models in 17 out of 21 categories. While the generalized model outperformed the category specific models in a majority of the categories, the category specific model (.676) had a higher f1 score than the generic model (.668). Ideally the next step would be to create an ensemble model that would use the generalized model to predict on the categories that it outperformed the category specific model and vice versa.

Conclusion:

I've found that there are patterns in how reviews are deemed helpful vs not helpful. There are several differences between the two groups.

- Reviews that reference the product with terms that pertain to the category specifically are found more helpful - i.e. musical terms such as sharp, flat and adagio have a higher probability of being helpful in the music category.
- Words with a bad probability of being helpful are those that pertain to a family member. These are reviews for products that were purchased for someone else (grandson, granddaughter, nephew).
- Words that are subjective and share the reviewers sentiment are much more likely to be found unhelpful than words that are objective which describe the product.
- For summaries in every category, the words denoting a number and "stars" showed up as bad predictors for helpfulness.
- Reviews that are more critical of the product are also found more helpful than those that praise the product.
- Older reviews have a higher rate of being helpful - Time has a dramatic effect on the helpfulness of reviews.
- Helpful reviews and summaries are longer than unhelpful reviews and summaries.
- A random forest model only using reviews as an input yields the highest f1 score.