Analysis of the Cell Phones & Accessories data set from Amazon

Data consists of about 79000 reviews.

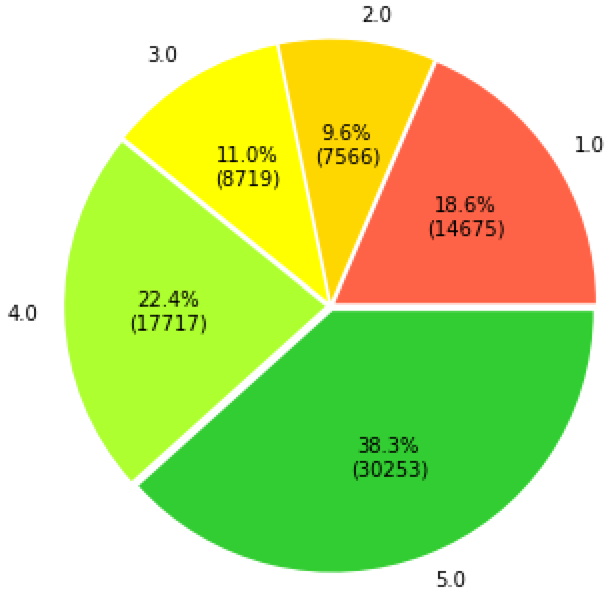
Each review is characterized by:

- **product/productId**: asin, e.g. amazon.com/dp/B00006HAXW
- **product/title**: title of the product
- **product/price**: price of the product
- **review/userId**: id of the user, e.g. A1RSDE90N6RSZF
- **review/profileName**: name of the user
- **review/helpfulnessFraction**: fraction of users who found the review helpful
- **review/score**: rating of the product
- **review/time**: time of the review (unix time)
- **review/summary**: review summary
- **review/text**: text of the review

I've also added two columns:
review/helpfulnessCount - number of user who found the review helpful
review/viewsCount - number of users who saw the review

More than half of the reviews refer to the product with an unknown price.
2276 reviews are created by unknown users

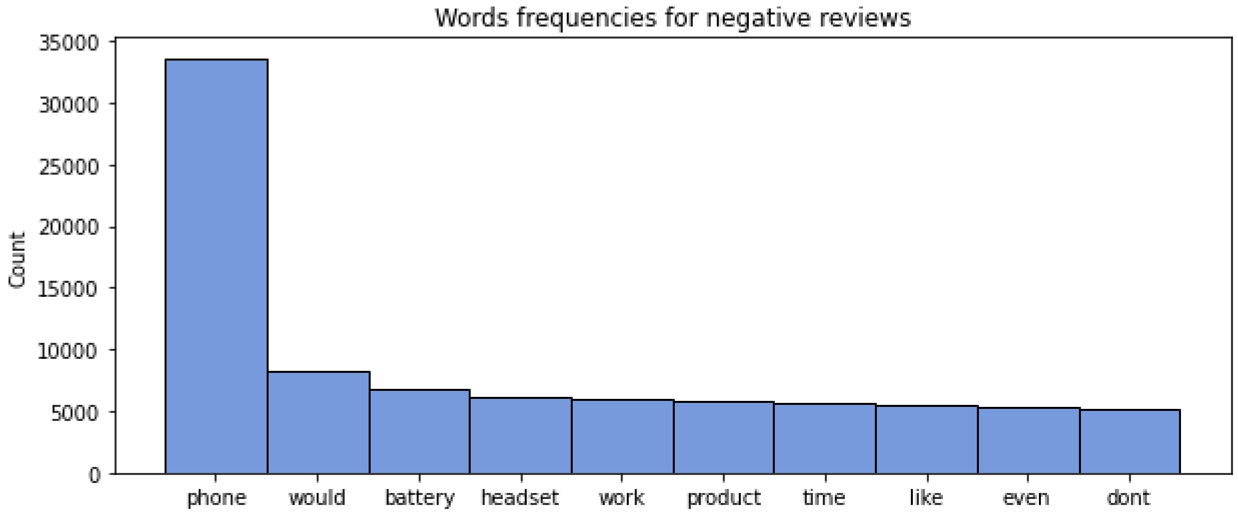
Looking at the distribution of the review scores, it's clear that customers are more likely to rate a product as being very good or very bad than something in the middle.

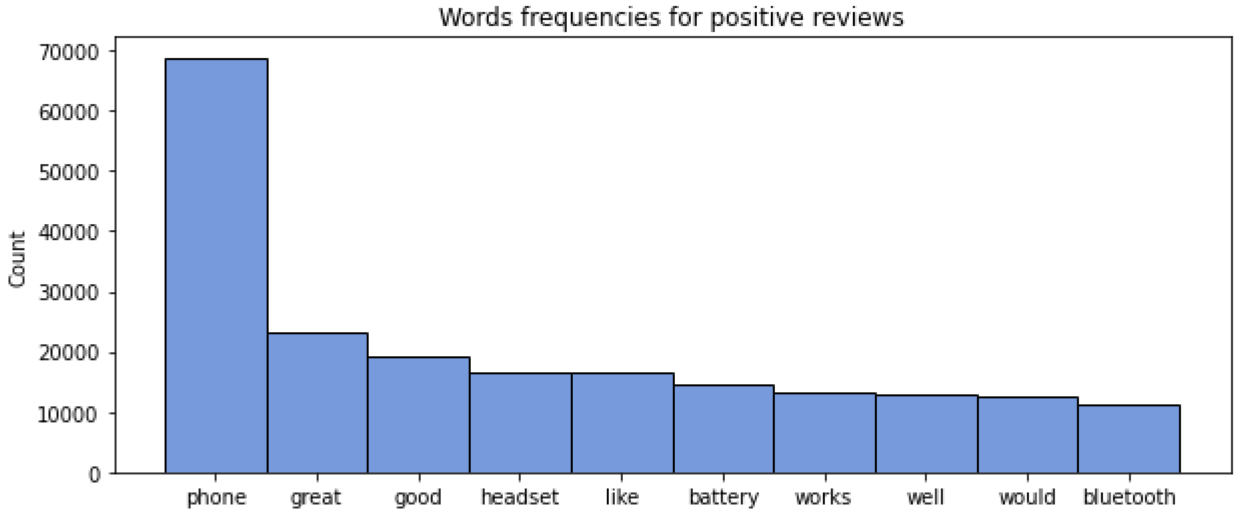


The top ten most frequently occurring words in positive vs negative reviews

I've decided to treat the score of 3/5 as a neutral review and henceforth don't count it as a positive or a negative. Before counting the occurrences of words, I cleaned the texts from stopwords.

Full reviews' texts:

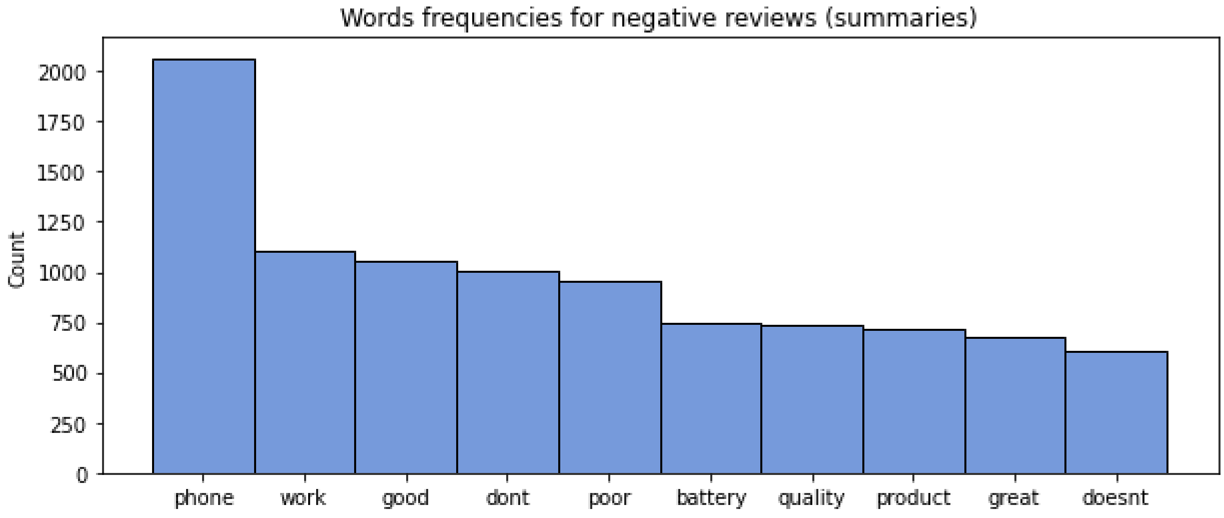


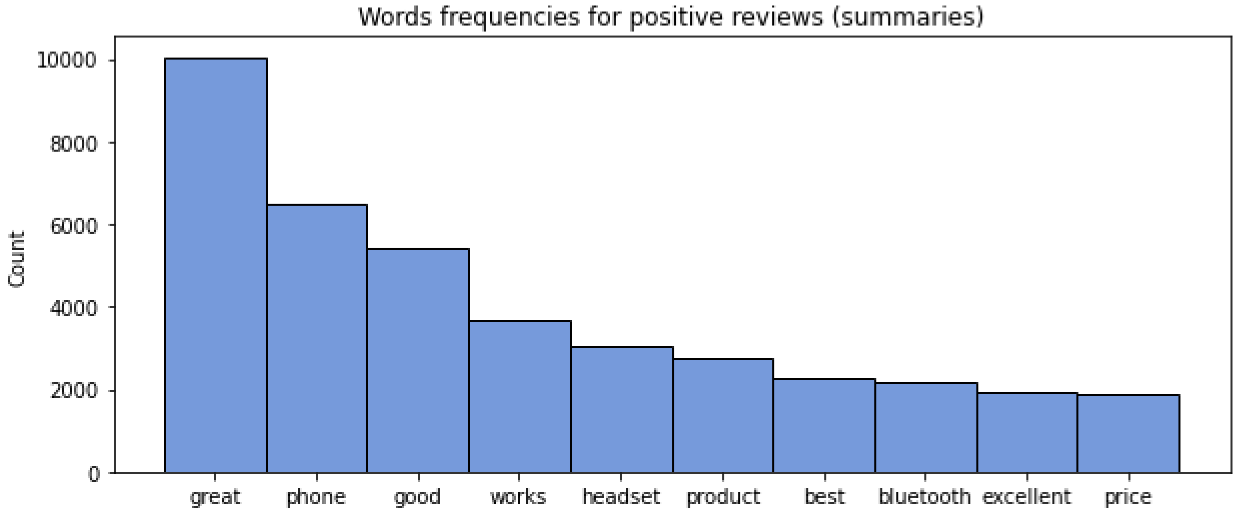


There aren't any adjectives among the most frequent words in the negative reviews. However, positive reviews contain a few of them - e.g. "good" or "great". Both of these groups are dominated by product names like "phone", "battery" or "headset".

The tenth most common word in negative reviews is 'dont' which probably comes from descriptions of malfunctioning products and customers expressing their dislike.

Reviews' summaries:





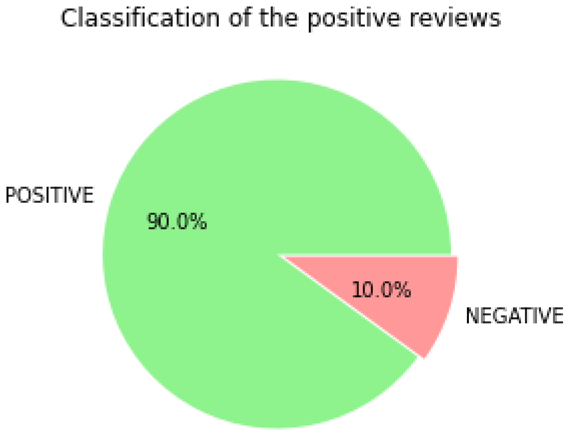
Based on the words frequencies, we can see that the summaries are more specific than the full reviews. Both negative and positive include many adjectives ('best', 'poor', 'good'). Product names are no longer so clearly dominant words.

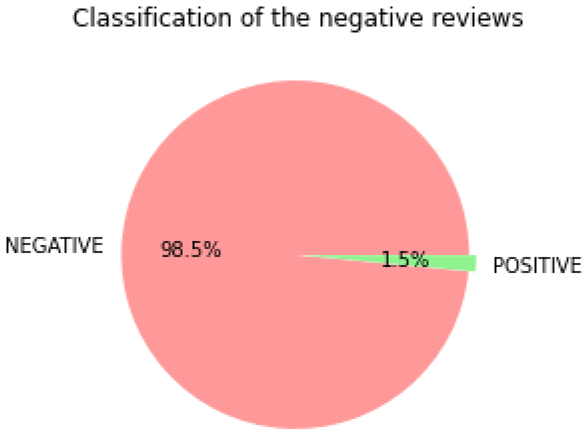
What exactly does a score of 3/5 mean?

It's obvious that in a five point rating scale one and two mean negative review, while four and five indicate positive review. However, I find three to be an ambiguous score.

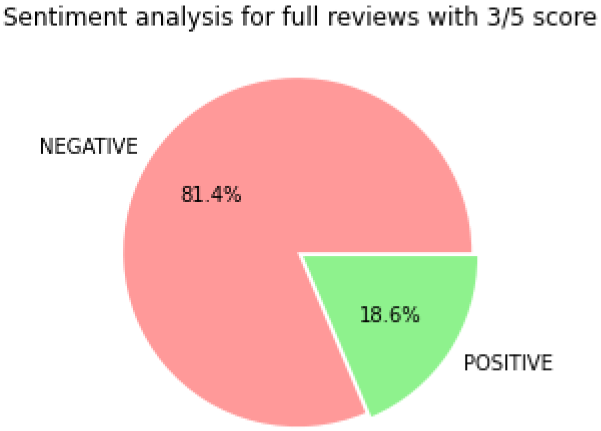
To check the actual nature of reviews with that scoring, I started by using a **pre-trained LSTM model** for sentiment analysis provided by flair framework. Since this neural net has been trained on IMDB reviews, the Amazon products reviews should not be a problem for it.

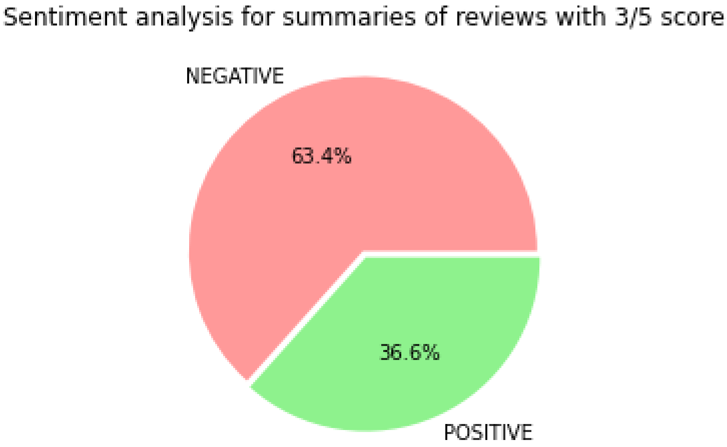
First thing to do, was to perform a sanity check of the model and see how it performs on a subsample of reviews with score 1 and 5.





With accuracy over 95% for the negative reviews and around 90% for the positive reviews, we can assume that the model performs decently on the dataset, and it makes sense to use it to classify reviews with a rating of 3. Besides the full texts, I also fed the model with the summaries.





According to the flair's model, **most of the full reviews** (~80%) with a score of 3 are in fact **negative**.

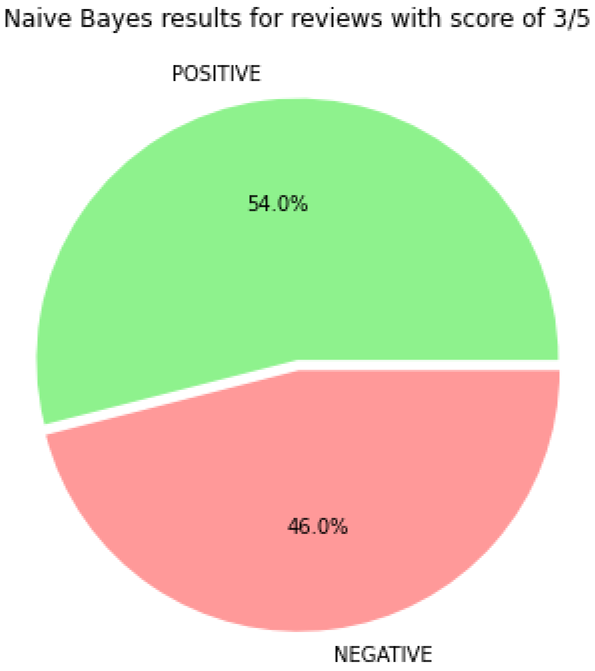
When considering the **summaries** of the reviews, the classification is a bit more ambiguous. About **62% of summaries are negative** and **38% is positive**.

Having in mind that the classifier is imperfect (only ~90% accuracy for the full reviews with 5 score), I cannot say without a doubt that negative summaries dominate over the positive when considering a score of 3/5.

The second thing I did was train the **Naive Bayes classifier** on the reviews with scores 1/5 and 5/5. I used a 0.7/0.3 train/test split and achieved the accuracy of around 90% for both positive and negative reviews (excluding the ones with scores 2/5 and 4/5).

I have artificially balanced the data used for this model. Before doing so, there were twice as many positive reviews as there were negative, which resulted in the Naive Bayes getting heavily biased towards classifying all of the text as positive (only 33.9% accuracy when testing on a set of negative reviews).

The results differed significantly from the results obtained by the LSTM model. According to Naive Bayes, there's a nearly perfect **50% / 50% split between positive and negative reviews** with a score of 3/5, which is another indicator that these reviews are not overwhelmingly positive or negative.

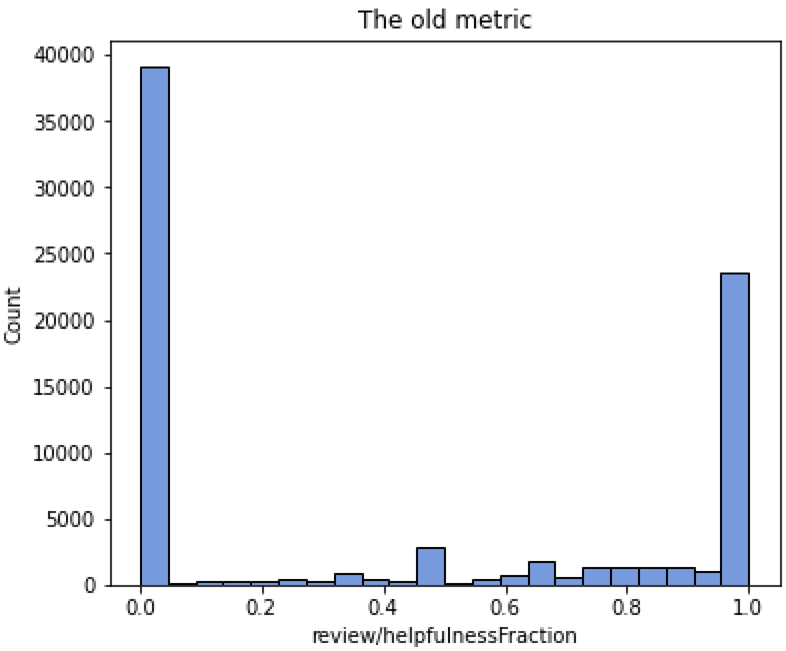


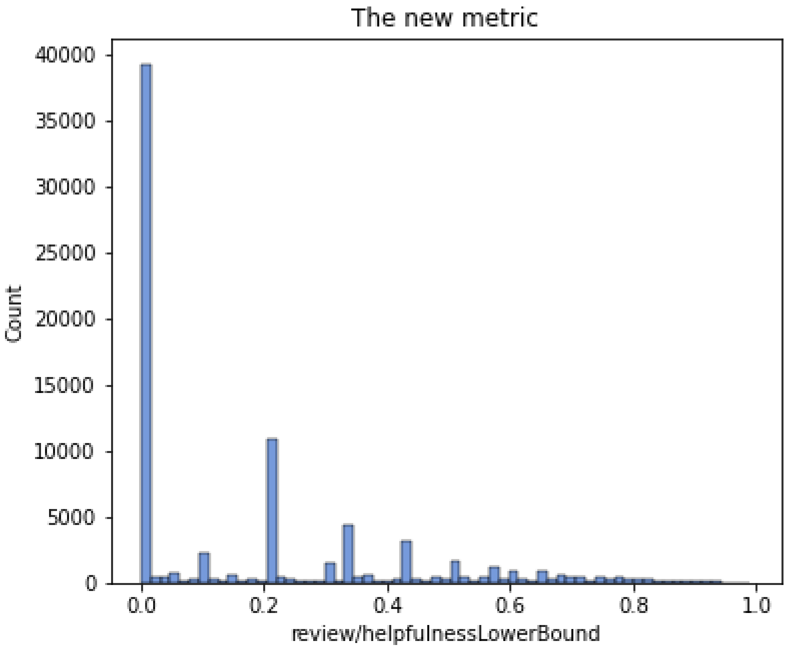
Creating a better metric for measuring the helpfulness of a review

Right now, the measurement of the review helpfulness is a fraction of people who liked the review. It's not very informative and can be misleading too - a review seen by only one person will be rated as perfect if this person liked it. It will even be counted as better than a review seen by 1000 customers, 999 of whom found it helpful.

To fix it, I will use a Lower bound of Wilson score confidence interval for a Bernoulli parameter (which is supposedly used by Reddit). It treats the review votes as a sample from a hypothetical set of votes made by all of the customers. Then, with 95% confidence, it calculates what would be the lower bound for a fraction of people who found the review helpful when considering the full customer population.

New reviews seen only by 1 or 2 customers won't be counted as very helpful since there's not enough data to confirm that. The more people will see the review and vote, the better the accuracy of the review helpfulness score will be.



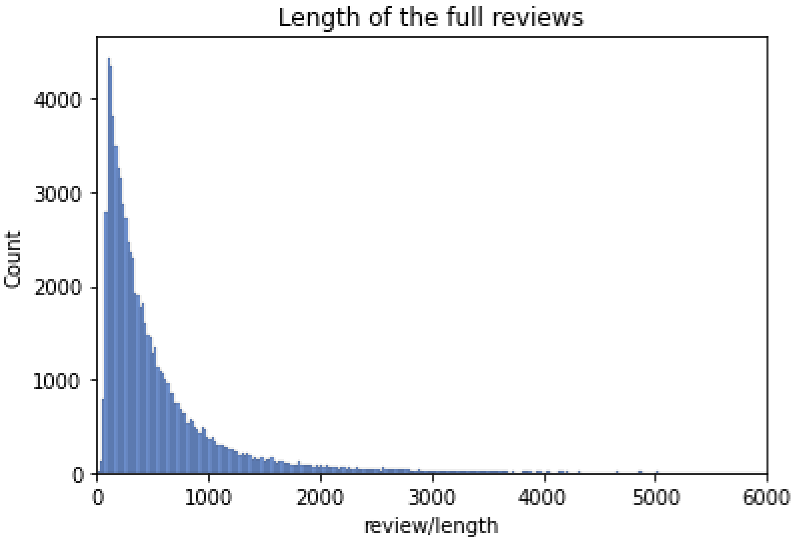


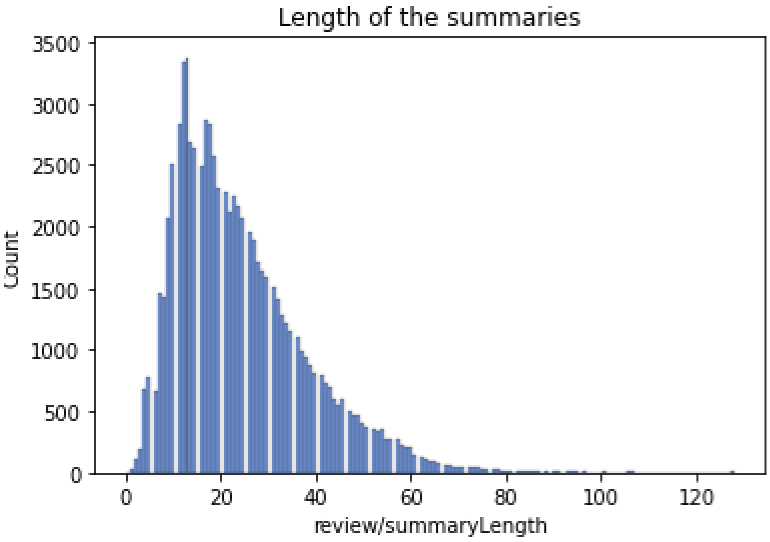
Based on the histograms above, we can see that the new metric (on the right) behaves much more naturally than the helpfulness measured as the fraction of people who liked the review (on the left). There is no sudden jump in the number of reviews as rating approaches 1.0. This jump occurs in the old metric and is caused by the reviews with very low number of votes.

Source that I used to find out how this metric works: https://www.evanmiller.org/how-not-to-sort-by-average-rating.html

Correlation between the measurements

With a proper helpfulness metric, I wanted to check if there are any correlations between the variables. Before doing so, for each review I counted the length of its full text as well as of the summary.



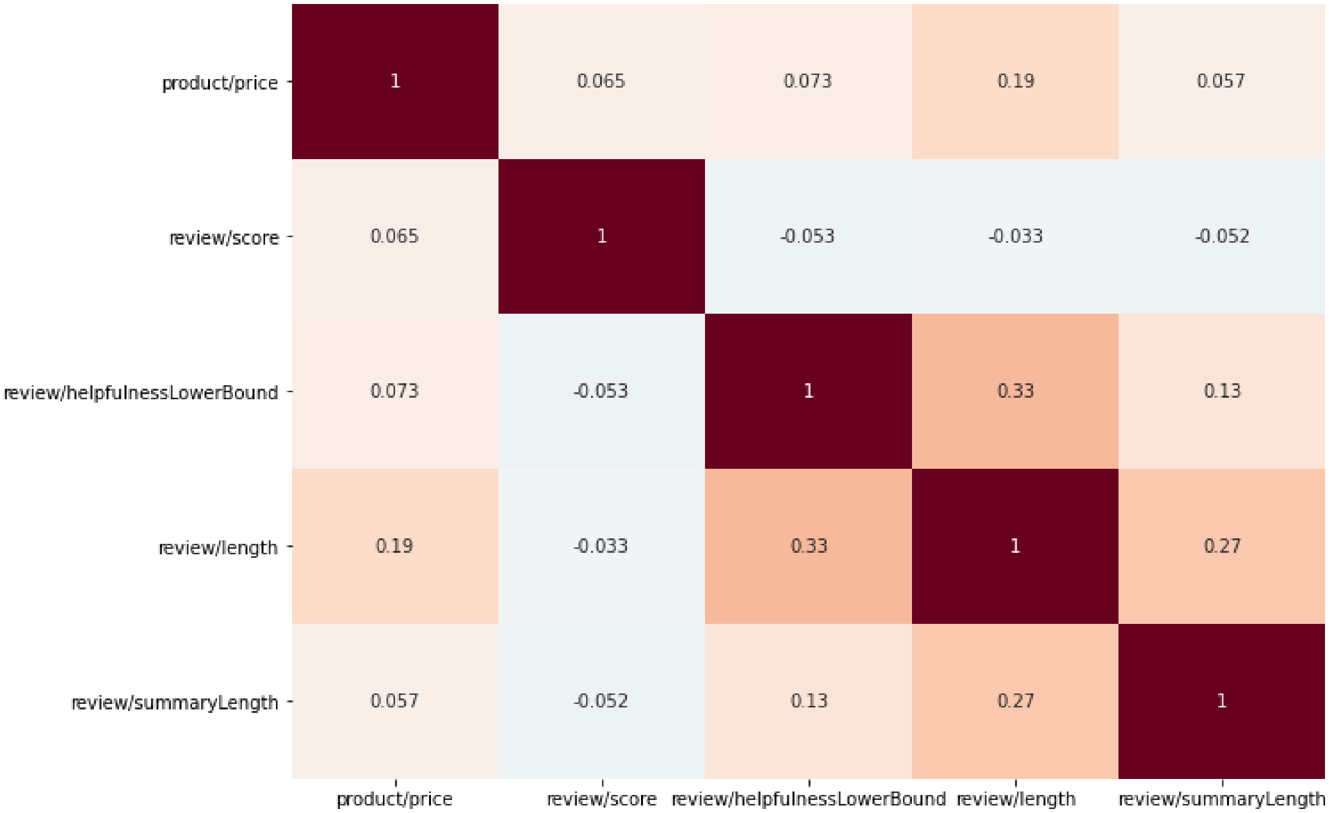


To calculate the correlations, I used the Pearson's coefficient.

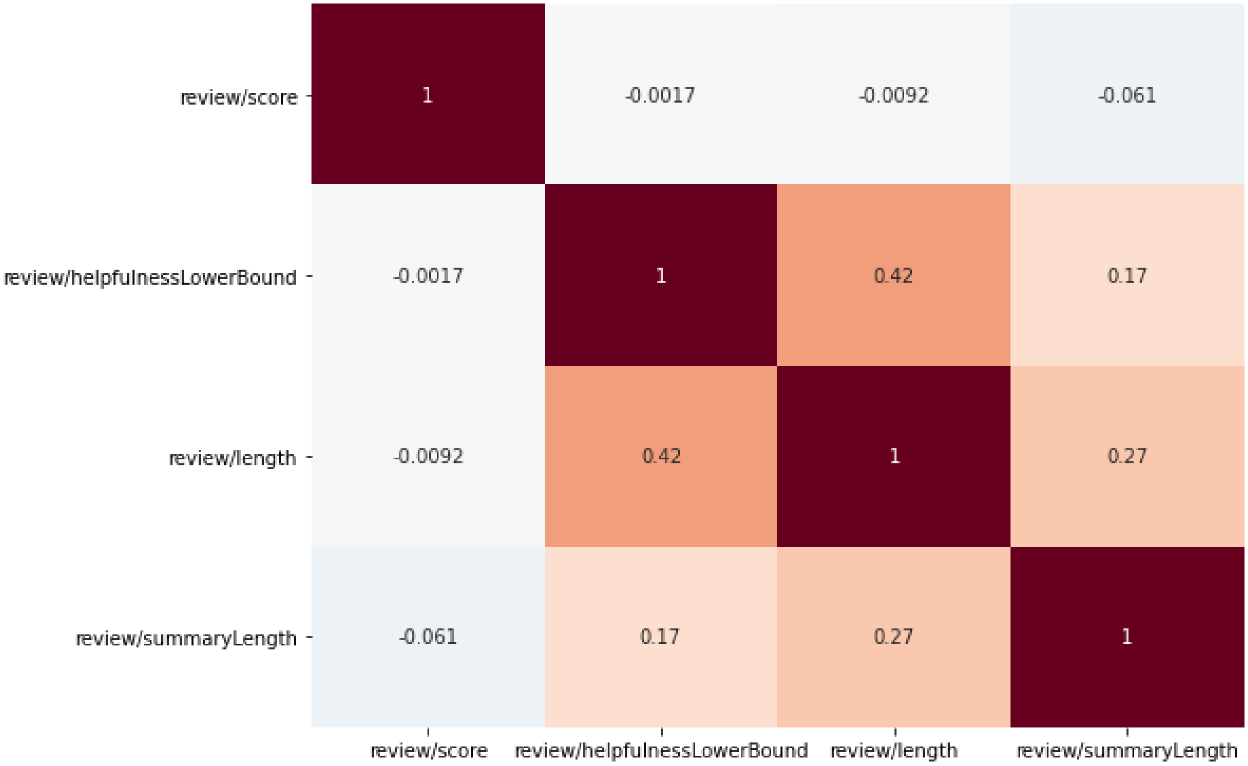
I've found three linear relationships:

- The obvious one - length of the full review is correlated with the review's summary's length (Pearson's coefficient of 0.27).
- There is a strong association between review's length and its helpfulness measured as a lower bound of Wilson score - the longer the review is, the more likely it is to be recognized as helpful. Correlation coefficient is 0.42.
- Price of a product is slightly correlated (0.19) with the product's review's length. This is a very self-explanatory relationship. If a product was expensive, people were more likely to spend extra time and effort writing a review.

Correlation heatmap for a data set cleaned of products with unknown prices



Correlation heatmap for a full data set



What could have been used for the recommendation system?

The algorithm I would like to try out is the Apriori algorithm. It has been widely used in the customers' cart analysis. It's based on two concepts: support (frequency of product occurrences) and confidence (how probable is product B to be purchased if product A has been bought). Having the transactions list, it builds subsets of products that meet the quality assumptions given by the support and confidence values.