Amazon US Customer Reviews

TINGHUI XU, OUYANG XU, BOWEN TIAN, YIJIN GUAN, YIFAN DU

Intoduction

Amazon is the online retailer with the largest variety of products in the world. It’ s meaningful for either customers or the business owners to know more about the reviews of the items. Our main goal is to explore which aspects are mostly mentioned. This can help sellers improve their stars. So we use CHTC to find the relationship between the high frequency words and rating stars.

Data Description

Source : Kaggle

https://www.kaggle.com/cynthiarempel/amazon-us-customer-reviews-dataset

Variable Names	Description
marketplace	2 letter country code of the marketplace where the review was written.
customer_id	Random identifier that can be used to aggregate reviews written by a single author.
review_id	The unique ID of the review.
productid	The unique Product ID the review pertains to. In the multilingual dataset the reviews for the same product in different countries can be grouped by the same productid.
product_parent	Random identifier that can be used to aggregate reviews for the same product.
product_title	Title of the product.
product_category	Broad product category that can be used to group reviews.
star_rating	The 1-5 star rating of the review.
helpful_votes	Number of helpful votes.
total_votes	Number of total votes the review received.
vine	Review was written as part of the Vine program.
verified_purchase	The review is on a verified purchase.
review_headline	The title of the review.
review_body	The review text.
review_date	The date the review was written.

Data Processing

- **split.sh**: to split all 6 tsv files into tens of small tsv files, each 100MB.
- **word_freq_array.sh**: to access these files to do the parallel computation for each category.
- **merge.sh**: merges all the csv files into a combined csv file using the given category.
- At last, we got **6 combined csv** files for further analysis, which were done in local.

Data Processing

Example:

- Split the camera.tsv (1.1GB) into 11 small tsv files.
- Run word_freq_array_camera.sh to launch 11 small jobs. Each job does the tokenization and lemmatization to each csv file, and then calculates the frequency of each word, returning a csv file with Column word, star_ratings, and frequency.
- A bash file named merge.sh merges all these csv files about camera into one csv file.

Text Preprocessing

- Tokenization on the customers' reviews.
- It's not so much about getting the correct answer, but how you get to the answer.
- Remove all of the punctuation and some html elements like
 and ".
- Turn each and every letter to lower-case letter and do lemmatization, which can transform words like 'swims', 'swam', 'swimming' to 'swim'.
- Do tokenization to separate the strings into single words.
- Remove all the stop words to get our final word boxes.

Computation for words frequency

- Groupe our word boxes to get words frequency results.
- Extracte top 2000 words in each split file and merge them into one .csv file.
- Select all the nouns from the words we get.
- Do these on each of the 6 sub files.

Computation for words frequency

Example Result

	product_category	star_rating	word.stem	frequency
13	Camera	star_1	accident	453
14	Camera	star_1	accommod	201
15	Camera	star_1	accompani	8
16	Camera	star_1	accomplish	26
17	Camera	star_1	account	1152

Word frequency v.s. Star rating plot

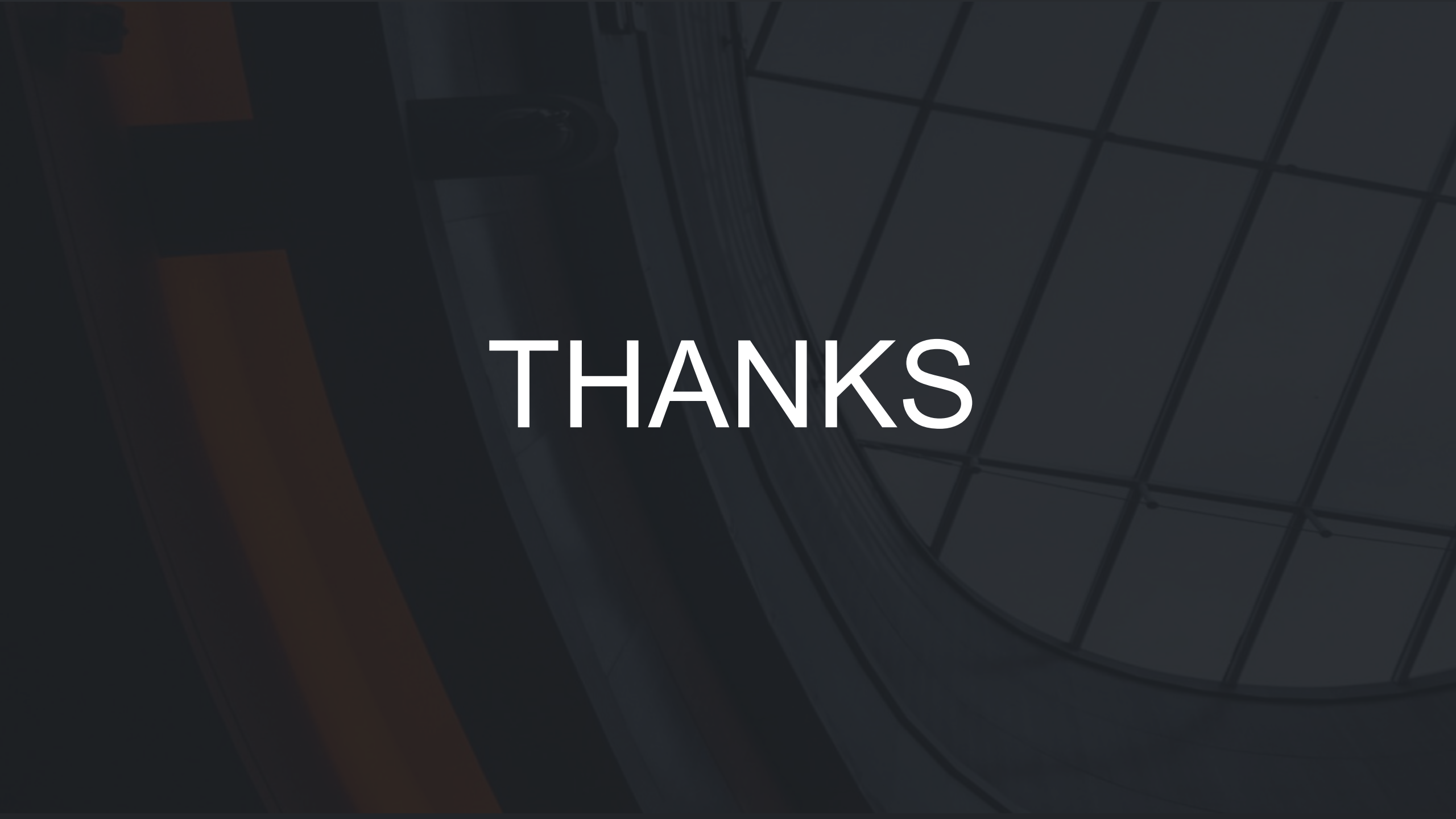
When we are computing words frequency, we keep the star rating and the product category of the reviews where certain words are split from. So we have the relationship between high frequency words and star ratings. And we can intuitively analyse some of the interesting words and based on these we can make suggestions or predictions.

Word frequency v.s. Star rating plot

Example Result

	word.stem	star_1	star_2	star_3	star_4	star_5
13	accident	453	391	786	2085	4200
14	accommod	201	188	457	1190	2203
15	accompani	8	4	10	43	72
16	accomplish	26	40	60	181	339
17	account	1152	563	855	1717	2731

Conclusion


THANKS