Amazon Product Review Analysis

Data 608: Project Proposal
(**Sneha Arora, Yongpeng Fu, Stuart Finley, Lu Li**)

Purpose

Amazon is a platform where we can find multiple brands for all types of products. It's a place that connects the buyer and the seller, where anyone can sell their product. Each product has a detailed information along with the ratings and reviews given by its customers. Users rely on this information provided to make relevant choices while buying a product of their choice.

The purpose of this project is to create a big data application for Amazon Headphone Products. The focus of our analysis is to use our web scrapped data to analyze what customers are saying about Headphone devices, discover insights into customer reviews, and potentially predict price based on reviews, ratings, and many distinct features like Volume control, Noise cancellation, and Connectivity technology. Eventually, we aim build a website interface to showcase this analysis.

Data

Web scraping tools – 'request' and 'beautiful soup' are used to extract *price, description, review date, customer reviews and ratings* for products from Amazon UK. This required developing a web scraper with multiprocessing capabilities to navigate the UK Amazon website and collect the desired data. As of Mar 14, 2023, we have over 200,000 reviews across ~200 products.

Web scrapping from Amazon is quite challenging - we are still working to improve our algorithms to collect more data. There are a few barriers to navigate with regards to pulling reviews from amazon, some of the most difficult being time requirements as well as denial of access. To help speed up the scraping process, the load was initially placed across all 4 of our members computers, with each member scraping a subset of the products reviews. While this distributed the load, page denial was still an issue. The cause for page denial is attributed to Amazon's efforts to block bots. For this reason, multiple user agents are generated randomly to help reduce the number of denials. To subsequently speed up the process of review scraping further, multiprocessing was leveraged.

A strength of this dataset is it is live data from the real world. We have control over the information we want from the website. As we approach our final analysis, we can feed our trained model with the most recent data.

Table 1: Following are the extracted columns used for our analysis, generated through web scraping

Product Name	The product names
Review Title	The title of each review
ASIN Number	Unique number given to each product
Price	The price of each item of the product
Description	Small description of each item given (for e.g., Wireless / in-ear headphones etc.)
Review date	The date when the customer wrote the review for an item (Each review has a different review date)
Customer reviews	Reviews written by the customer about how they like/dislike the product.
Customer Rating	Rating given by the customer out of 5 for a particular item, giving a quick idea of the overall performance of the product from their perspective.

Methodology

Web scraping

A generalized web scraper to continues to be developed to retrieve contents from Amazon website. Initial scraping will be conducted to retrieve links, price and the number of ratings for items returned from a general product search of ‘*headphone*s’. This scraping generates a list of links which are then modified to redirect to the ‘*reviews*’ page for that item. These links are then passed to another scraper which attempts to pull all of the reviews for the item in question. The process of pulling reviews proved to be challenging and slow, so multiprocessing was leveraged at this stage to increase speed. The final dataset generated from this stage in scraping includes the unique product identifier, the name, the title of the review, the body of the review, the rating as well as the date for which the review was given.

AWS Cloud Services

CodeCommit is utilized as our git repository. We will create a project branch from the ‘main’ branch, and then every team member will create their own branch, and at the end, we will merge our branches to the project branch.

We are using AWS RDS MySQL as our database which will be comprised of two tables: Product and Review.

We are following Agile Software Development. Each team member has their own responsibilities, and we have constant communication in the team to report what we doing and what problems we are facing.

Sentiment Analysis

Data cleaning and wrangling are required to remove the stop words, blanks and other information not required in the review data scraped from Amazon website. Sentiment analysis performed on the on the reviews of the products to derive the review rating used for price prediction, review summarisation as well as sentiment score to be compared with rating.

Predicting the price of a new item for a business user or a customer will be done using the features above in addition to the review rating and summary.

Dash Website Interface

To show our analysis and expose our general-purposed web scrapper algorithm, we will use Dash to build a simple one-page web application. We will use dash bootstrap components, and Dah Mantine components to build the layout and add interactive components (joined effort between teams). The final application environment will be stored in a docker container to be deployed in the Amazon EC2.

Individual Learning Objectives

Yongpeng Fu

- Incorporate multiprocessing technique in the web scrapper by Mar 27th.
- I will have a better understanding on HTML, CSS, Flask framework, and build the Dash web foundation by March 27th.
- Deploy our Plotly/Dash web using Docker and Amazon EC2 by April 1st.

Stuart Finley

- Develop web scraping abilities to pull various information types such as text, URLs and images using beautifulsoup by April 1st and use these skills to pull data for multiple different products and across multiple pages.
- Explore different avenues to perform sentiment analysis by examining the benefits and drawbacks of supervised vs. Unsupervised learning. Determine which method is most suitable for our set of data and establish sentiment scores by April 1st.

Lu Li

- Learn AWS services what used in our project and make sure all the services are running successfully by March 23.
- Learn Dash Python framework, and able to create the website for our project by March 27.

Sneha Arora

- Understand sentimental analysis and create review summarization using Abstractive summarization technique with spacy for each item of the product by 15 March.
- Generate review rating using sentiment analysis and compare with the customer rating. Find the difference to give a realistic picture told by the reviews by 1 April.
- Price prediction using the summary and review rating by 7 April.

Feedback Received (Pitch)

Good work with just some minor points for improvement.
I really like that there are clear indications that you're already working creating your system and working to understand what will work and what won't.

Also remember to keep an eye on minor details - the technical term is "sentiment analysis" rather than "sentimental analysis." I think the typefaces you used for your slides changed a little bit throughout your pitch - always helpful to do a quick check for these little things!
The project seems reasonably sized. Good work!

References

- Gangwar, M. (2022) *How to scrape Amazon product information using beautiful soup*, *DigitalOcean*. DigitalOcean. Available at: https://www.digitalocean.com/community/tutorials/scrape-amazon-product-information-beautiful-soup (Accessed: March 2023).
- *Fake-useragent* (no date) *PyPI*. Available at: https://pypi.org/project/fake-useragent/ (Accessed: March 2023).
- Jhnwr, J. (2020) *Scrape-amazon-reviews/review-scraper.py at main · JHNWR/Scrape-Amazon-reviews*, *GitHub*. Available at: https://github.com/jhnwr/scrape-amazon-reviews/blob/main/review-scraper.py (Accessed: March 2023).
- Bhatt, A., Patel, A., Chheda, H. and Gawande, K., 2015. Amazon review classification and sentiment analysis. *International Journal of Computer Science and Information Technologies*, 6(6), pp.5107-5110.
- Sblendorio, D. (2022) *How to do text summarization with Deep Learning and python*, *ActiveState*. Available at: https://www.activestate.com/blog/how-to-do-text-summarization-with-python/ (Accessed: March 2023).