WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

PRICE PREDICTION AND DYNAMIC PRICING IN E-COMMERCE

MOHAMMED ARSHATH S[1], Ms.V.PRIYADHARSHINI[2]

[1] III-BSC.CS-"B", DEPARTMENT OF COMPUTER SCIENCE SRI KRISHNA ADITHYA COLLEGE OF ARTS AND SCIENCE, KOVAIPUDUR,COIMBATORE
[2] ASSISTANT PROFESSOR DEPARTMENT OF COMPUTER SCIENCE SRI KRISHNA ADITHYA COLLEGE OF ARTS AND SCIENCE, KOVAIPUDUR,COIMBATORE

ABSTRACT :

With the rapid growth of e-commerce and advancements in technology, the number of online shopping platforms has increased significantly, offering door-to-door services to customers. However, the abundance of websites makes it difficult for users to find quality products at affordable prices. This application provides a user-centric solution by retrieving product prices from multiple online stores and presenting the best options to the user. Additionally, it tracks the user's purchasing history to offer personalized product recommendations. The system also includes an intuitive user interface and a chatbot to assist with FAQs and website navigation, thus saving time and enhancing the overall user experience.

Keywords: Web Scraping, HTTP, Chatbot, Visual Analysis, Recommendation

I.INTRODUCTION :

In recent years, online shopping has experienced remarkable growth, driven by increased internet accessibility and the rise of e-commerce platforms. As consumers face a vast selection of products across numerous websites, finding the best price for a desired item has become a time-consuming challenge. This paper presents an innovative shopping platform that utilizes web scraping techniques to streamline this process, helping users quickly identify the best prices available across multiple e-commerce sites. The system is designed to monitor price fluctuations and notify users of any price drops in real-time. To enhance the shopping experience, a chatbot powered by machine learning assists with seamless navigation and answers frequently asked questions.

Moreover, the platform includes personalized features by analyzing the user's purchasing history. This allows the system to generate tailored recommendations, which helps users manage their budgets and track their shopping habits. The remainder of the paper is structured as follows: Section 2 discusses related works in this domain. Section 3 outlines various systems studied. Section 4 presents the usability analysis,
while Section 5 describes the algorithms employed. Finally, Section 6 provides the results, followed by the conclusion.

2. RELATED WORK :

Various methods related to product recommendations, web scraping, and chatbots have been explored, as these features significantly enhance the usability and efficiency of shopping platforms, improving the overall user experience.

2.1 Personalized Recommendation System

Personalized recommendation systems not only improve customer satisfaction by offering relevant suggestions but also assist sellers in boosting their revenue and enhancing the platform's overall quality. Several types of recommendation systems have been studied, including content-based, rule-based, and collaborative-filtering approaches. It has been observed that incorporating semantic techniques into recommendation systems can lead to more accurate suggestions [1]. In e-commerce, multi-agent systems (MAPRS) have been used to build recommendation engines by assigning different tasks to various agents. However, this approach has certain limitations, such as being static and having potential security vulnerabilities [2].

2.2 Chatbot

A chatbot is a virtual assistant designed to help users with navigation and respond to common inquiries, thereby improving customer satisfaction and reducing the need for a dedicated support team for minor queries. Research has highlighted that chatbots should be well-trained using natural language processing (NLP) techniques to handle more complex queries effectively [3]. Additionally, it is essential for chatbots to have a simple and user-friendly interface to facilitate seamless interactions.

2.3 Web Mining

Web scraping and web crawling are techniques used to extract data from web pages, and they have been employed by various researchers for different purposes [4]. These methods, such as using Python's requests, Beautiful Soup, and Selenium, have proven effective in retrieving e-commerce deals from multiple websites with high accuracy [5].

3. SYSTEM STUDY :

We have examined two existing systems to understand their functionalities and limitations, which we aim to address and improve in our project. The first system, widely used around the world, is the "Google Shopping" platform, designed to help users find products across various online stores. The second system, "Pennywise," is explored in the research paper [5], which compares products across four different e-commerce websites.

3.1 Google Shopping System

Launched by Google in December 2002, Google Shopping was designed to provide a centralized platform for users to find products from different vendors online. Users can enter a search query, browse categories, and view products on sale [6]. The system ranks products from various shopping sites based on relevance, incorporating search terms and other Google activity. Product information is updated regularly, providing users with real-time details from sellers. The system's key features include [7]:

1. **Powerful**: Google Shopping leverages Google's search technology to help users discover and compare products from online retailers worldwide.
2. **Comprehensive**: It offers a wide range of products, from common items to rare finds.
3. **Efficient**: Google Shopping quickly displays related products with images and links to detailed information, including the stores selling them.

However, there are some drawbacks to this system. While Google Shopping does provide a variety of product options from different websites, results marked as "Sponsored" are ranked based on the advertiser's payments to Google, potentially influencing the recommendations users see [7]. This could hinder users from receiving unbiased results. In our system, we aim to address this issue by ensuring that products are ranked purely by relevance to the search, without any influence from advertisements.

Additionally, Google Shopping tracks a user's purchase history to offer personalized product recommendations. However, it does not provide a clear way for users to view or analyze their purchase history. Our system seeks to improve upon this by offering a more visual comparison of products using graphical analysis tools, along with displaying the user's purchase history in an accessible and user-friendly format.

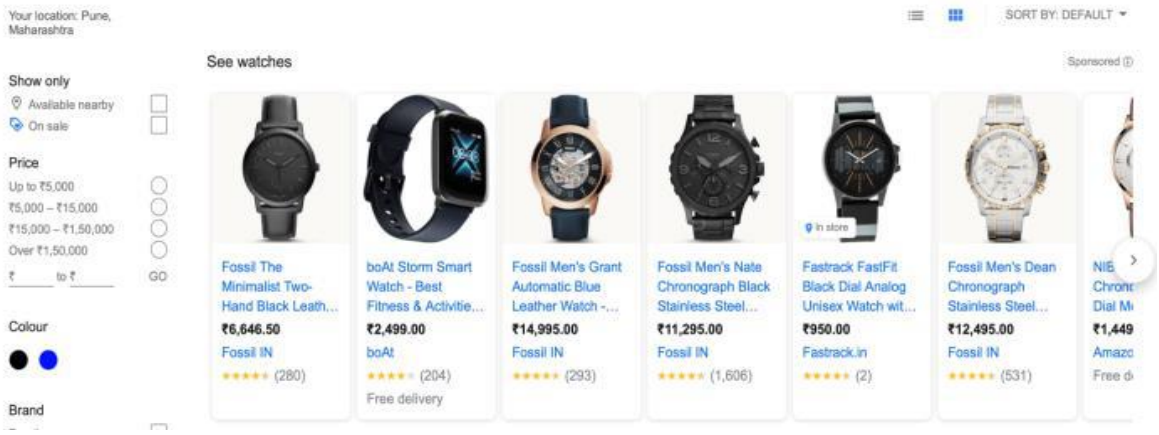
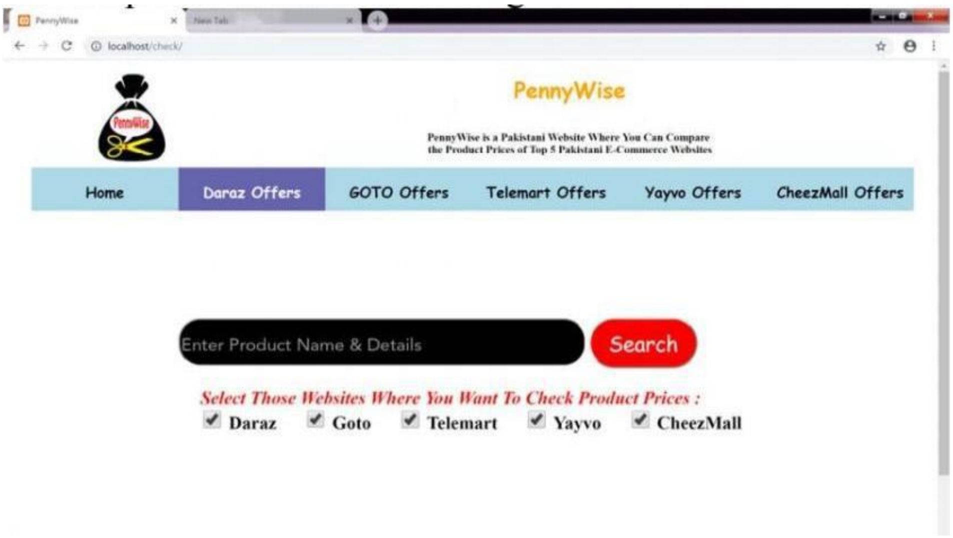

Your location: Pune, Maharashtra
SORT BY: DEFAULT
See watches
Sponsored
Show only
Available nearby □
On sale □
Price
Up to ₹5,000 ○
₹5,000 – ₹15,000 ○
₹15,000 – ₹1,50,000 ○
Over ₹1,50,000 ○
₹ to ₹ GO
Colour
Brand
Fossil The Minimalist Two-Hand Black Leath... ₹6,646.50 Fossil IN (280)
boAt Storm Smart Watch - Best Fitness & Activitie... ₹2,499.00 boAt (204) Free delivery
Fossil Men's Grant Automatic Blue Leather Watch -... ₹14,995.00 Fossil IN (293)
Fossil Men's Nate Chronograph Black Stainless Steel... ₹11,295.00 Fossil IN (1,606)
In store
Fastrack FastFit Black Dial Analog Unisex Watch wit... ₹950.00 Fastrack.in (2)
Fossil Men's Dean Chronograph Stainless Steel... ₹12,495.00 Fossil IN (531)
NIE Chron Dial M ₹1,449 Amazo Free d

Fig -1: Google Shopping System

3.2 Pennywise System

The Pennywise system, as described in the research paper [5], compares products across five different websites and presents the results to users for efficient shopping. While it provides quick and accurate results, it has several limitations that our system aims to address. One major drawback is that the Pennywise system does not offer sorting or filtering options for the results, making it difficult for users to refine their search based on specific preferences. In contrast, our system incorporates a filtering feature that allows users to customize results according to their budget and preferences.



PennyWise

PennyWise is a Pakistani Website Where You Can Compare the Product Prices of Top 5 Pakistani E-Commerce Websites

Select Those Websites Where You Want To Check Product Prices :

☑ Daraz ☑ Goto ☑ Telemart ☑ Yayvo ☑ CheezMall

Fig -2: Pennywise System

Additionally, the Pennywise system is purely search-based and does not track users' interests or purchase history. To improve this, our system integrates a user-based application that not only monitors users' past purchases but also analyzes them to offer personalized product recommendations through a machine learning recommendation system. Furthermore, our platform enables users to track products they are interested in and receive notifications when those items drop below a price threshold they have set. This feature, powered by our Notification System, helps users stay updated on price changes and find the best deals.

SYSTEM ARCHITECTURE :

The system is designed with three key components: the Price Comparer, the Chatbot, and the Analyzer. The Price Comparer and the Analyzer are integrated into a single module, while the Chatbot operates as a standalone module. Together, these components work to fulfill all the functional requirements of the application.

The database is user-based and hosted on a cloud machine, accessible through Firebase. All modules interact with this database to perform their specific tasks. The Price Comparer module focuses on retrieving product prices from various websites to offer the best price. It also monitors the user's preferred products, which are stored in a separate user-based Wishlist database.

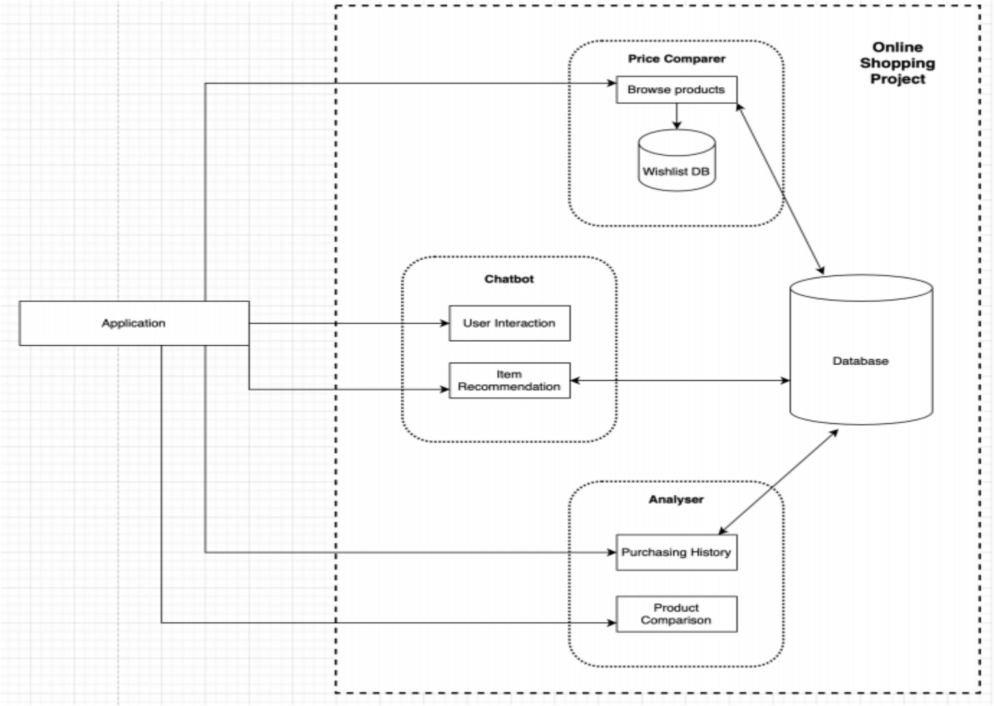


Fig -3: System Architecture

4.1 Analyzer Module

The Analyzer module is responsible for comparing and extracting prices for different products across various websites. It uses web scraping tools like Selenium and Beautiful Soup in Python to collect data. This module also tracks user preferences by categorizing products, enabling it to recommend similar items that the user is likely to be interested in. These recommendations are based on a user-category database [8][9]. Additionally, the module has to access .The Analyzer module also tracks the user's purchase history, which is presented in graphical form to help the user better analyze their buying patterns.

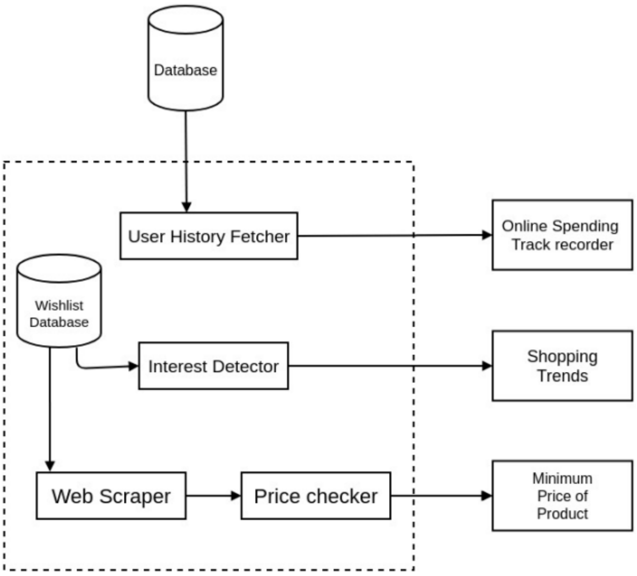


Fig -4: Analyser Design

4.2 Chatbot Module

The Chatbot module is a text-based interface designed to assist users by answering their queries and guiding them within the application. It also displays product recommendations generated by the Analyzer module, helping users find relevant products based on their interests and previous interactions.

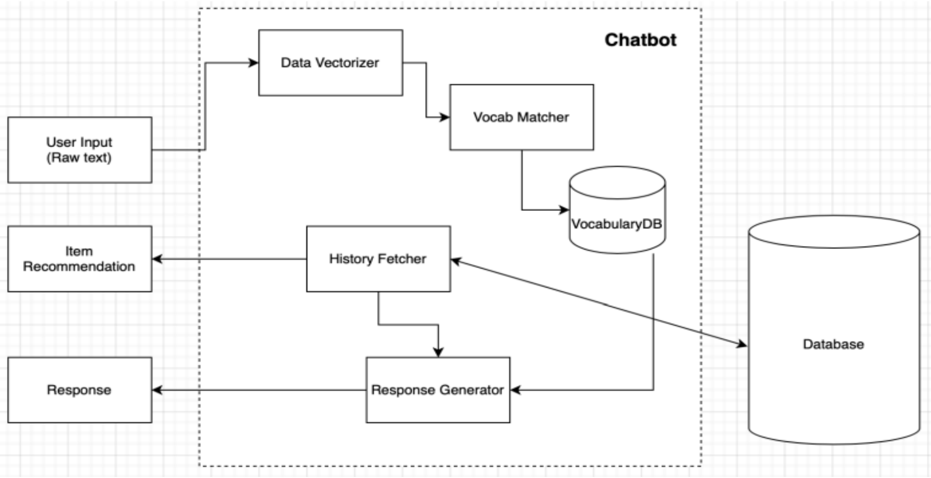


Fig -5: Chatbot Working

This module begins by processing the user's text input, matching it with the chatbot’s vocabulary. It then classifies the input into various categories, each corresponding to predefined responses. Additionally, the module can access the user database to retrieve the user's purchase records.

5. ALGORITHM :

5.1 Chatbot:

The Chatbot is designed to improve the user experience and make the website more intuitive and user-friendly. It aims to facilitate seamless interaction by responding to user queries and guiding them through the platform.
The Chatbot is designed to assist users with frequently asked questions (FAQ) and provide basic navigation across the website.

Algorithm 1: Chatbot Algorithm

1. **Creating the Training Dataset**

1.1. A categorical dataset is created, grouping similar messages or questions into specific categories.
1.2. Responses are defined for each category to guide the chatbot's replies.

2. **Pre-processing the Training Data**

2.1. All messages/questions are tokenized and categorized accordingly.
2.2. Words are stemmed to retrieve their root forms.
2.3. Punctuation marks are removed from the data.
2.4. Redundant words are eliminated using the "Sets" function.
2.5. The cleaned data consists of a list of words for each category.

3. **Training the Machine Learning Model**

3.1. A deep neural network with two hidden layers and a "softmax" activation function is used.
3.2. The training data is used to train the model and optimize the weights for each connection in the network.

4. **Processing User Inputs**

4.1. User input is tokenized and stemmed to extract root words.
4.2. The input is compared against the list of words to generate a one-hot encoded list.
4.3. The list is then fed into the neural network.
4.4. The output consists of probabilities, showing how similar the input is to each category.
4.5. The chatbot responds with the message from the category with the highest probability.

5.2 Wishlist and Purchase History Algorithm

The Wishlist is a personalized data table stored in an online database, which keeps track of the user's desired products and their corresponding websites. Similarly, the purchase history is also stored and used to analyze buying patterns via graphical tools.

Algorithm 2: Wishlist / History Algorithm

- Display results of user search queries from multiple websites.
- Direct the user to Wishlist or History upon clicking "Add to Wishlist" or "Buy Now."
- Verify the existence of a user-specific Wishlist/History table in the database.
- If the table doesn't exist, create a new table for the user.
- Retrieve all items from the user's Wishlist/History.
- Check if the product is already present in the user's table.
- 5.1. If it exists, skip to Step 7.
- Add the product with relevant details (website, URL, title) to the database.
- Display all items in the Wishlist/History table for the user.
- For purchase history, provide a graphical analysis of the user's monthly spending and cost breakdown by product.

5.3 Notification System Algorithm

The user can set up price drop alerts by adding products to a notification list. The system will notify the user when the price of the product meets or falls below the user-defined threshold.

Algorithm 3: Notification System

1. Select a product, set a price threshold, and add it to the notification list.
2. Append the product and threshold to the list in the database.
3. Repeat Steps 1 and 2 to add additional products.
4. Check real-time prices for products in the notification list using their URLs with Beautiful Soup.
5. Store the fetched prices in the database to track trends.
6. Notify the user via email if a product's price falls below the threshold.
7. Repeat for all users and products.
8. Automate the entire process.

5.4 Recommendation System Algorithm

The recommendation system employs a collaborative filtering approach, which uses preferences gathered from users to predict interest in products. Products are categorized based on user purchase history, and the nearest neighbor algorithm maps these relationships to make recommendations.

Algorithm 4: Recommendation System Algorithm

1. Clean the dataset by removing records with missing values and irrelevant columns (e.g., 'user id').
2. Categorize each product into one of seven product categories and add this information to the dataset.
3. Split the dataset into a training set (80%) and a test set (20%) for model development.

5.4 K-Nearest Neighbors (KNN) Model

The K-Nearest Neighbors (KNN) model is employed to make product recommendations based on user history. Here's a breakdown of how the model works:

1. The KNN model is trained and tested using the dataset.
2. For each user, their historical data is accessed to retrieve relevant product information.
3. This data is then used to predict and generate product recommendations from the trained KNN model.
4. The system displays the recommended products to the user.

5.5 Web Scraping

Web scraping is the process of extracting product information from various online shopping websites. This is achieved through a series of steps:

Algorithm 5: Web Scraping Process

1. Take the product name as input for searching across different websites.
2. Pass the product name in URL format to web scraping tools like BeautifulSoup and Selenium.
3. Extract the product label, price, image, and hyperlink for each product.
4. Compare the extracted product prices to match them with the search query.
5. Display the products in ascending order of price, filtered for accuracy.

For Flipkart Website

Flipkart uses asynchronous loading techniques, making it difficult to scrape data using BeautifulSoup alone. To handle this, Selenium is used to retrieve product information even when content is loaded dynamically.

Algorithm 6: Flipkart Scraping Process

1. Pass the product name to the Selenium driver for search.
2. Scrape the product label, price, and image using the XPath and class attributes of the web elements.
3. For specific product types, adjust the scraping process by using different XPath or web element identifiers.
4. Append the extracted product details into a list.
5. Repeat the process for all displayed products on the page.
6. Return the compiled list of product details.

The algorithm accommodates different product categories by adapting the method for extracting details based on the type of product being viewed.

6. Results and Discussions :

The proposed system enables users to easily compare prices of a product across various e-commerce websites. The system displays the top search results in a unified interface, allowing users to visualize prices and choose the best deal.

The average time taken by the system to fetch and display price information for a product from multiple websites is 5.8 seconds, significantly reducing the time users would spend manually searching and comparing prices. Additionally, the notification system alerts users about price drops for products they are interested in, providing notifications via email when the price falls below a set threshold. This feature enhances the user experience by saving time and reducing the effort required to monitor product prices.

7. Conclusion :

The Online Shopping Platform offers an efficient way for users to find the best prices for products across various websites. By leveraging advanced web scraping techniques and analytics, the system accurately determines the best price and presents it to the user, avoiding the need to visit multiple e-commerce sites.

The platform's consistent user interface simplifies the shopping experience, making it easy to navigate and find the best deals. Additionally, the integrated notification system ensures that users are promptly informed of any price changes for their preferred products, eliminating the need to track prices manually across different platforms.

The personalized recommendation system enhances the user experience by suggesting relevant products based on their shopping history. This, coupled with detailed analytics of the user's purchase history, enables better management of shopping budgets and habits. Furthermore, the chatbot, powered by machine learning algorithms, streamlines navigation, assisting users in easily finding products and features within the platform.

Overall, the system serves as a comprehensive solution for frequent online shoppers, consolidating various features such as price comparison, product recommendations, price tracking, and user-friendly navigation into one unified platform.

REFERENCES :

[1] Shaikh, M., & Rathi, D. S. (2017). Recommendation system in E-commerce websites: A Graph-Based Approach. *IEEE 7th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*.

[2] Guo, Y., & Liu, Q. (2010). E-commerce Personalized Recommendation System Based on Multi-Agent. *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010)*.

[3] Galitsky, B., & Ilvovsky, D. (2017). Chatbot with a Discourse Structure-Driven Dialogue Management. *Proceedings of the EACL 2017 Software Demonstrations*, Valencia, Spain, April 3-7 2017, 87–90.

[4] Julian, L. R., & Natalia, F. (2015). The use of web scraping in computer parts and assembly price comparison. *2015 3rd International Conference on New Media*.

[5] Mehak, S., Zafar, R., Aslam, S., & Bhatti, S. M. (2019). Exploiting Filtering Approach with Web Scraping for Smart Online Shopping. *2019 International Conference on Computing, Mathematics, and Engineering Technologies (iCoMET 2019)*.

[6] Wikipedia contributors. (n.d.). *Google Shopping*. Retrieved from https://en.wikipedia.org/wiki/Google_Shopping

[7] Google. (2002). *Google Shopping*. Retrieved from https://support.google.com/faqs/answer/2987537?hl=en

[8] McAuley, J. (2018). Amazon product data. Retrieved from http://jmcauley.ucsd.edu/data/amazon/links.html

[9] Ni, J., Li, J., & McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[10] Zhang, Y. (2020). *Introduction to Collaborative Filtering*. Towards Data Science. Retrieved from https://towardsdatascience.com