International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Online Shopping Analysis and Product Price Comparison Using Web Mining and Machine Learning

Akash Kumar[1], Sanyam Saklecha[2], Shreyas Pawar[3], Vaibhav Kumar[4], Prof. N.A. Mhetre[5]

[1]*BE Student, Department of Computer Engineering, Singhad College Of Engineering,Pune, Maharashtra, India*
[2]*BE Student, Department of Computer Engineering, Singhad College Of Engineering,Pune, Maharashtra, India*
[3]*BE Student, Department of Computer Engineering, Singhad College Of Engineering,Pune, Maharashtra, India*
[4]*BE Student, Department of Computer Engineering, Singhad College Of Engineering,Pune, Maharashtra, India*
[5]*Professor, Department of Computer Engineering, Singhad College Of Engineering,Pune, Maharashtra, India*

Abstract - *With the increasing technology and rapid growth in e-commerce nowadays, a number of online shopping websites are seen, which provide door-to-door service to the end user. But, due to the vast number of websites, the user has to spend a lot of time to find quality products at a cheap price. The given application will contain a user-based system, which retrieves the costs of the same product across various different websites, providing the best choice to the user, along with keeping track of the user's purchase history, to recommend products the user is most likely to buy. Hence, saving time and, at the same time, providing a good and friendly user interface along with a chatbot, to answer FAQs and help with navigation across the website.*

***Key Words*: Web Scraping, HTTP, Chatbot, Visual analysis, Recommendation.**

1. INTRODUCTION

Over the last few years, online shopping has seen a tremendous amount of growth. As the Internet is becoming more and more accessible worldwide, the number of online buyers and e-commerce platforms are rapidly increasing. With a vast variety of products to choose, from a plethora of websites, the user is inundated with choices, when it comes to choosing the best price of a product available online. A considerable amount of time and efforts are spent when the product one needs is available at different e-commerce sites. In this paper, we demonstrate a shopping platform which leverages powerful web scrapping techniques and incredible features to save the user's time and unnecessary drudgery one has to go through while finding the best price for a product. Our system continuously checks for the price updates and provides the user with the best price for a product in real-time. A notification system is also employed which communicates the user about a price-drop whenever it occurs. Navigating through multiple platforms while shopping online can be sometimes unintuitive, confusing and time- consuming. In our system, navigation is made seamless by the Chatbot developed using Machine Learning algorithms. It assists the user in navigating through the platform and provides a smooth user experience while shopping. In addition, we have a personalized recommendation sys- tem which improves the shopping experience by suggesting products to the user based on their interests. Our system uses analytics tools which keep track of the user's purchasing history and provide a detailed analysis to the user, which helps the user in maintaining their shopping budget and assess their shopping habits. The rest of the paper is organized as follows: Section 2 introduces related work in the same field. The different systems studied and reviewed are described in Section 3. The usability analysis is presented in Section 4 following with a section on algorithms. Results are discussed in Section 6 before concluding with the paper.

2. RELATED WORK

Several methods related to recommendation of products, web scrapping and chabot were reviewed. These features improves the usability and efficiency of a shopping platform along with enhancing the shopping experience of a user.

2.1 A personalized recommendation system

Personalized recommendation system not only satisfies the customers by offering relevant recommendations, but also helps the seller to increase their revenue and improve the overall quality of the shopping platform. Different types of recommendation systems were studied, including Contentbased, Rule-based and Collaborative-filtering recommendation systems. We found that integration of semantic in recommendation techniques can provide better recommendation [1]. The recommendation system for e-commerce can be created using multi-agent doing different tasks, referred as MAPRS, but the system has particular drawbacks as it is static and has security issues [2].

2.2 Chatbot

A computer program that is available as a virtual assistant for the customer to help them in navigation and answering common questions, which improves customer satisfaction and saves the cost of having a support team for small queries. Various papers stressed that a chatbot should be well-trained using NLP for answering long and complex queries[3] and should have a non-complex interface for interaction.

2.3 Web mining

Web-scrapping and web-crawling are the techniques of extraction of data from web pages. These techniques were implemented by various researchers for different purposes [4]. Scrapping a web page using python request, Beautiful Soup, Selenium provides high accuracy in providing the best ecommerce deals for customers from different websites [5].

3. SYSTEM STUDY

We have studied two different systems for their workings and shortcomings which we aim to simplify and conquer in our project. The first system is the globally used "Google Shopping" system used by users all over the world to quickly find their products on various websites online while the other system, "Pennywise", is studied from the approach given by the research paper [5] which searches for products across four different websites.

3.1 Google Shopping System

This system was launched by Google in December, 2002 which aimed at providing a commonplace for all the products online on different vendors. The user can enter a search query and look for individual products, browse departments, and see the products which are on sale [6]. It ranks various products offered online on various shopping sites which are then ranked based on relevance, including your search terms and other Google activity. Users can access product information from sellers that is updated on a regular basis via Google Shopping. The most powerful features of this system are [7]

1) Powerful: Google Shopping is a powerful tool that uses Google's search technology to help you locate and study items from online retailers all over the web before directing you to where you can purchase them.
2) All-inclusive: You'll find a wide range of items, from the everyday to the rare.
3) Robust: You'll see images of related items and links to more detail, including the stores that sell them, almost immediately after entering a search.

Nevertheless, there are certain aspects to this system which might not always be beneficial to the user. The system does indeed provide results with a variety of products across different websites, but, the results of the system under the "Sponsored" label are ranked and grouped according to their advertisement payment to Google, as per their policy [7]. This can be a hinderance to the user and also influence the user's purchasing choices. We aim to avoid this problem in our system, by providing unbiased results and ranking

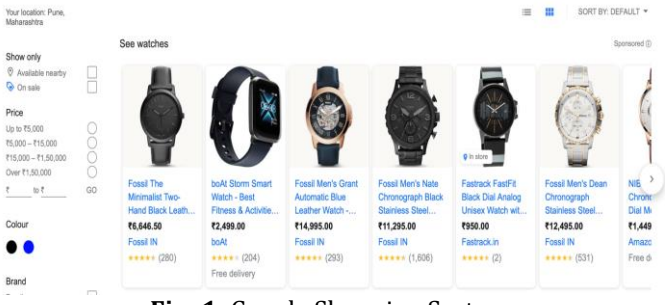

Fig -1: Google Shopping System

products purely based on relevance with the search. Although, Google keeps track of the products purchased by a user to provide personalized product recommendations which we aim to include in our system, the Google Shopping system does not provide the user a way to analyze and view their purchasing history records. Our system works to provide a more visual comparison of products to the user with the help of graphical analysis tools along with displaying the user's purchase history in the form of user-friendly graphs.

3.2 Pennywise System

This system was proposed in the research paper [5], which compares products across five different websites and provides the results to the user to shop effectively. The system is able to provide the results quickly and efficiently. The overall accuracy of the system is also decent.

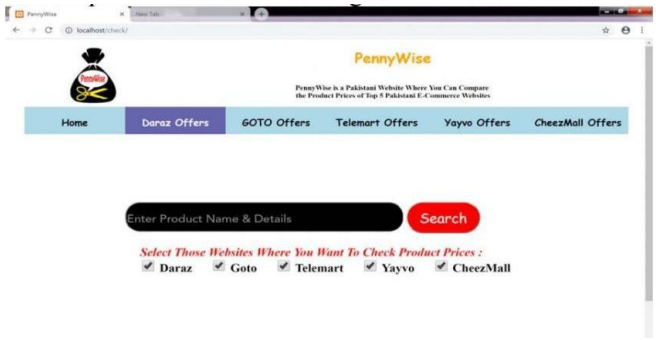

Fig -2: Pennywise System

However, there are quite a few limitations to this system which we focus on eliminating in our system. The Pennywise system does provide results which are accurate, but, there is no way of sorting and filtering the results displayed. Our system, implements the filtering approach thus providing the results better suiting to the user's budget and choices. The Pennywise system is also purely search based system which does not keep track of the user's interests and purchases. Our system includes these features by including a user-based application which keeps track of the user's purchase records to not only provide analysis of the purchasing history, but to also recommend products using a Machine Learning Recommendation System. Furthermore, our system also lets user's track their interested products and have the choice of being notified when a product falls

below a certain price threshold set by the user, to allow the user to wait for the perfect price of products. This is implemented using our Notification System which allows the user to easily keep track of the prices of their interested products.

4. SYSTEM ARCHITECTURE

The system consists of three different components namely the Price Comparer, The Chatbot and An Analyser, which carry out all the functional requirements of our application. The two components of the price comparer and the analyser are clubbed into a single module, while the chatbot serves as an independent module.

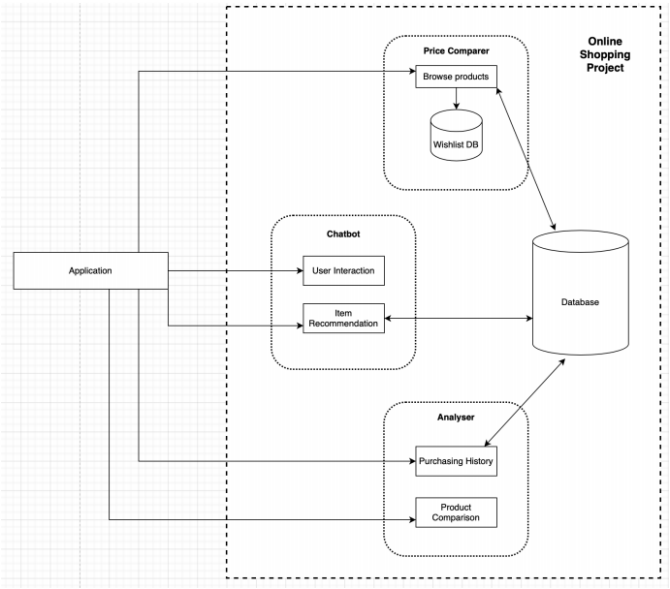

Fig -3: System Architecture

The database consists of a user based database which is run online on a cloud machine accessed by the firebase database. All of the modules have access to this database to each carry out their independent functioning. The price comparer is a simple module which aims to mine the prices across various websites to provide the best price along with keeping track of the user's interested products using the independent user based Wishlist database.

4.1 Analyser Module

This module is concerned with the comparison and mining of the different prices for the various products across websites, to be displayed to the user, using python web mining tools of Selenium and Beautiful Soup. The module also keeps track of the user's interests in the form of product categories which is used to provide recommendation of similar products, which the user is most likely to buy based on a user-category database [8][9]. Along with this, the module has access to the

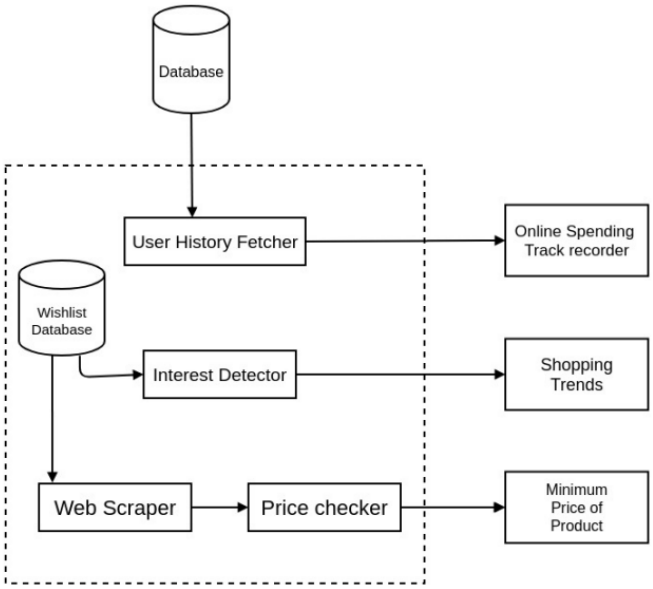

Fig -4: Analyser Design

user's purchase record, which is then used to display the user's purchase record graphically to better analyze the data.

4.2 Chatbot Module

The Chatbot module is a text based module which helps answering user queries in the form of text by redirecting the user within the application, along with displaying the results of the user's product recommendations obtained from the Analyser module.

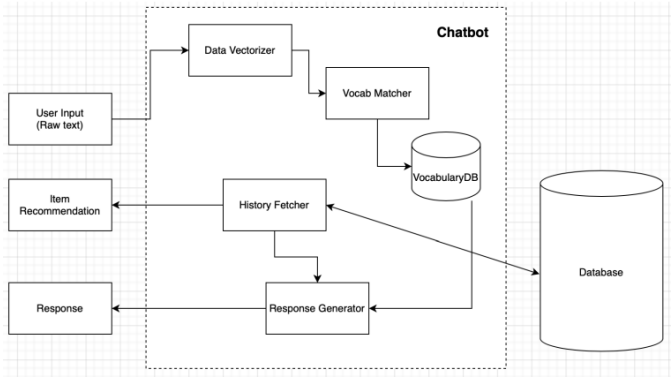

Fig -5: Chatbot Working

This module first extracts the user's text input to match with the vocabulary of the chatbot, which then classifies into several categories which contain pre-defined responses for the users. This module also has access to the user's database for fetching the purchase records of the user.

5. ALGORITHM

5.1 Chatbot

Chatbot is implemented with an aim to enhance user's experience and make the website more user-friendly. It is

used for answering FAQ(frequently asked question) and also providing basic navigation across the website.

Algorithm 1: Chatbot Algorithm

1 Creating the training dataset
  1.1. Training dataset is a categorical dataset which groups similar messages/questions into a single group.
  1.2. Adding responses that bot is expected to give for a particular group.
2 Pre-processing of training data
  2.1 All messages/questions are tokenized from each group and stored into a list based on each group.
  2.2. Stemming of all words from the list is done to get root word for each of them.
  2.3. Any sort of punctuation symbols are removed.
  2.4."Sets" is used to remove redundant words from the list.
  2.5. Training data is obtained that consists of words list for each group.
3 Training of machine learning model
  3.1. Machine learning model used is a deep neural network that consists of two hidden layers and "softmax" as activation function.
  3.2. Training data is passed as the input to train the model and get optimized value of weights for each link in the neural network.
4 Processing user inputs
  4.1. User input is also tokenized and stemmed to get root words out of the sentence.
  4.2. It is then, compared with all words list to check which words are entered by user and create one-hot encoded list for the same.
  4.3. The resultant list is then, sent as input to the neural network.
  4.4. The output is the probabilities of question's similarity with each of the groups 4.5. The response of the chatbot will b selected from the group that has the highest probability among the all.

5.2 Wishlist and History

The Wishlist is a user specific data table implemented using an online database, which keeps saves all of the user's interested products along with their websites, which the user can then browse through at any given point, while the product history for the user is implemented in a similar fashion which keeps track of all the user bought products which can be analyzed with the help of graphical tools.

Algorithm 2: Wishlist / History Algorithm

1 Display results of the search query of the user from all the websites.
2 Redirect to Wishlist / History functions on button click of "Add to Wishlist" / "Buy now".
3 Check if user specific Wishlist / History tables exist for the user.
  3.1 If yes, create Wishlist / History table in the database for the particular user.
4 Retrieve all the items in the tables for that user.
5 Check if the product already exists in the tables for that user.
  5.1 If yes, continue from Step 7.
6 Add the product along with all the other attributes like website, URL, product title in the database.
7 Redirect to display page to display all the items in the Wishlist / History table for the user.
8 In the case of history display, provide a graphical analysis to display monthly expenditure of the user along with product wise graphs with cost.

5.3 Notification

User will add products to into the notification list and the notification system would notify the user whenever the price drops or meets the threshold price of that product.

Algorithm 3: Notification System

1 Select the product, enter a threshold price for it and add to the notification list for a user.
2 Append the product to the notification list along with its threshold price and hyperlink.
3 Repeat the Steps 1 and 2, to add more products into notification list.
4 For each product present in notification list check the price using the hyperlinks in real-time using Beautiful Soup
5 Store the fetched price in the database to keep track of price trends.
6 If the price of product is found to be less then threshold price, then the user is notified via email.
7 Repeat Steps 3 to 6 for all users.
8 Automate the whole script.

5.4 Recommendation System

The Recommendation system is based on the collaborative filtering approach which gathers preferences information from a large number of users to predict a user's interest on a product [10]. The product interests are based on categorical basis of products which are taken from the purchase history of the user, and are mapped to those with a similar pattern based on users from the dataset [8][9]. The Nearest Neighbor Algorithm is used to map this relationship to recommend products to the user.

Algorithm 4: Recommendation System Algorithm

1 The dataset is cleaned by removing records with Nan values and trimming off columns like 'user id'.
2 Each product details are then mapped onto one of the seven categories and added into the dataset.
3 The dataset is then shuffled and cut into a 80-20 ratio for the train-test development of the model.

4 K Nearest Neighbor model is trained and tested on the dataset.
5 For every user, the history tables are accessed to retrieve the product information of the users.
6 This track record of the user is used to predict and produce recommended products from the KNN model.
7 The results are displayed to the user.

5.5 Web Scrapping

Algorithm 5: Web Scrapping Algorithm

1 Taking product name as input which is to be searched across different websites.
2 Passing the product name in the form of URL to Beautiful Soup and Selenium.
3 Scrap product label, price, image and hyperlink for every product from each website.
4 Compare the product price and match them to the search query.
5 Display the website in ascending order of the product's price while being filtered on accuracy basis.

1) For Flipkart website: Flipkart uses asynchronous loading techniques for showing products so it becomes difficult to get the data using BeautifulSoup. That's why an alternative, Selenium, is used which allows to scrap data, even when the data to be scrapped from is loaded using asynchronous techniques or Ajax.

Algorithm 6: Flipkart Algorithm

1 Pass the name of the product to be searched is to Selenium driver.
2 Scrap the product label, price and image using the web elements' XPath and class.
3 If the product belong to a type of distinctive cases, then for each special case, the product details are scraped using the respective web-elements' XPath.
4 Append the product details in a list.
5 Repeat Steps 2 and 3 for each product displayed on page.
6 Return the list of product details.

The products are differentiated in the way they are displayed on the basis of their categories. Hence for distinct types of products, the web element which stores product details is changed. To include such change in web elements, distinctive cases have been made to make the algorithm functional.

6. RESULT AND DISCUSSIONS

The proposed system compares and displays the prices of a product from different e-commerce websites. Top search results are displayed to the user on a single interface. Visualization of product prices helps the user in determining the best price for a product. The system redirects the user to the original website of a particular product, in case the user chooses to buy any product.

The average time taken by a user to manually search and compare price of a product on different websites is quite high, whereas the time taken by our system to fetch the price of product on the same websites is 5.8 seconds on average. So, the user can get the best price of a product within 5 seconds. Also, the user gets notified about the price drop of an interested product by the notification system via email. Hence, our application saves time and inconvenience caused to user while searching and comparing prices of products online.

7. CONCLUSION

The Online Shopping Platform enables the user to find the best price available of a product on the internet with ease. As there are many available e-commerce websites with thousands of products, our system uses powerful web scrapping technologies and analytics to determine the best price of a product. The consistent application interface helps user to avoid unnecessary hassle to navigate through different websites while looking for best available price. Due to unification of search results from various platforms at a single destination, it becomes incredibly easy to use this platform. The integrated Notification System in our application provides a much coherent point of access and enables user to keep track of the prices of their favorite products. The Notification System communicates the user about the price drop whenever it occurs. Henceforth, reducing the inconvenience of signing up into different websites just to get notified for a price drop. The Online Shopping Platform not only determines the best product price but it also allows for an improved personal shopping experience by suggesting different products to users using the Personalized Recommendation system. The Analytics keeps track of a user purchasing history and provide a detailed analysis which helps user in maintaining their shopping budget and assess their shopping habits. Navigation through multiple platforms while shopping online is many times unintuitive and time-consuming. So, in our application navigation is made seamless by the Chatbot developed using Machine Learning algorithms. It assists the user in navigating through the platform and provides a smooth user experience while shopping. So, our system is able to reduce time and efforts in searching for the best price for a product from different websites. When it comes to comparing prices of a product, product recommendation and price tracking of a product, with the vast numbers of features available in our system, it becomes a one-stop solution for frequent online shoppers.

REFERENCES

[1] Ms. Shakila Shaikh , Dr. Sheetal Rathi, "Recommendation system in E-commerce websites: A Graph Based Approached", 2017 IEEE 7th

[2] GUO Yan-yan, LIU Qi-cheng, "E-commerce Personalized Recommendation System Based on Multi-Agent", 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).

[3] Boris Galitsky, Dmitry Ilvovsky, "Chatbot with a Discourse Structure-Driven Dialogue Management", Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017, pages 87–90.

[4] Leo Rizky Julian, Friska Natalia, "The use of web scraping in computer parts and assembly price comparison", 2015 3rd International Conference on New Media.

[5] Shakra Mehak, Rabia Zafar, Sharaz Aslam, Sohail Masood Bhatti, "Exploiting Filtering approach with Web Scrapping for Smart Online Shopping", 2019 International Conference on Computing, Mathematics and Engineering Technologies – iCoMET 2019.

[6] Google shopping, wekepedia, https://en.wikipedia.org/wiki/Google Shopping

[7] Google Shopping, Google, Accessed December 2002 https://support.google.com/faqs/answer/2987537?hl=en

[8] Julian McAuley, UCSD, "Amazon product data" 2018 , http://jmcauley.ucsd.edu/data/amazon/links.html

[9] Jianmo Ni, Jiacheng Li, Julian McAuley, "Justifying recommendations using distantly- labeled reviews and fined-grained aspects", Empirical Methods in Natural Language Processing (EMNLP), 2019.

[10] Yuefeng Zhang, PhD, Published on 5 August 2020, https://towardsdatascience.com/machine-learning-for-buildingrecommender-system-in-python-9e4922dd7e97