International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 03 | Mar 2026 www.irjet.net p-ISSN: 2395-0072

Cost Curve – Dynamic Price Tracker using Web Scrapping

Sandip Shinde[1], Aakarshit Gupta[2], Aayush Wase[3], Aditya Shah[4], Aditya Chauhan[5]

[1]*HOD, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India*
[2]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India*
[3]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India*
[4]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India*
[5]*Student, Dept. of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India*

Abstract - *E-commerce is quite popular, in the recent years the field has had a growing demand for some sort of a tool which helps a user identify a product and track/compare its prices across different websites (Amazon, Flipkart, etc.), so that the user can make educated decisions regarding their purchase. Cost Curve does just that – this paper presents the algorithm and the need of a tool which tracks prices in real time. The tracker makes use of web-scraping techniques to feed the pricing data from a number of e-commerce platforms and these prices are fetched into a user-friendly interface where the user can either—look at the prices across websites, history of the prices over a specific period of time or set a threshold at which the user wants to buy the product. If the product reaches a price below said threshold, the user is notified about it so that they can take advantage of the discount and save money. Overall, Cost Curve is a valuable tool for the customers to chase deals and prices, all at the right time.*

Key Words: **Web scrapping, Dynamic Pricing, E-Commerce, Price Comparison, Price Alerts, Data-Driven pricing**

1. INTRODUCTION

E-Commerce, Online shopping, and all these technologies have enabled an individual to place an order in the comfort of his or her home. Over the past few years, an increasing number of individuals have been choosing to buy products and services online. As a result of this, the online market has turned into a very competitive one and has made a large imprint on the local businesses. It has compelled local businesses to offer the best of prices and present other forms of deals at the appropriate time besides other innovative actions aimed at attracting and retaining the customer.

An example of how an offline business can match with the online market is by checking the price of a particular product and then setting the price of his/her products based on the online market this also allows the businessman to give discounts at the appropriate time. This assists the business in making a pricing decision that is data-driven.

The online price tracker is not only efficient to a local business-it also helps a customer to investigate the inflated/ anomalous price of a particular platform in order to know whether it would be appropriate to buy a product on the specific platform or the other platform. He is also able to check the product price history, to ensure that the user is able to make his purchase at the appropriate time, and not be fooled/deceived by the price. This helps the customer to make a wise purchase that would end up saving his or her money.

2. LITERATURE REVIEW

Dynamic pricing adopted by e-commerce platforms has made price monitoring and analysis a necessity. A number of studies have examined systems which monitor prices, predict trends or help consumers to locate those purchase windows which would be the most advantageous to them. An example is the Parashar et al. [4] paper where they deployed a fundamental Product Price Tracker, which gathers price information of the online stores and stores them in a database that users can query. Also, the Pilakavil et al. [2] designed a Real-Time Product Price Monitoring and Analysis system that assists a customer in establishing the most suitable buy-price prior to making an online purchase. These primitive systems defined the need to have historical records of prices and display it in a manner that is useful in informative shopping.

Moreover, the approaches of mobile-based monitoring in terms of technology have been investigated, including Price Watch application by Cayabyab et al. [3], which used Exponential Smoothing to predict prices of agricultural products and showed the utility of the integration of prediction-based functions to the user-oriented system. Khairnar et al. [1] took this direction a step further by incorporating dynamic pricing algorithms in an e-commerce price tracker demonstrating how algorithmic logic and web scraping can be used to present competitive prices in a more efficient manner. All these works emphasize on the necessity of some automated, data-driven systems that should be able to analyze current and past price data.

Pricing strategies and decision-making frameworks have also been the issue of heavy research. Nowak and

Pawlowska-Nowak [5] investigated the use of machine learning models of SVM, Naive Bayes and KNN to assist in managerial price adjustment decision-making. Once Xu et al. [6] and Miao et al. [7] presented reinforcement learning and clustering-based models, respectively, they demonstrated the advanced AI-driven methods to optimize the pricing under unpredictable or sparse-data conditions. Even though each of them leans towards the sellers, not the consumers, each of such studies demonstrates how hard price changes can get on the e-commerce sites, an issue that is simplified by the tracking software (such as CostCurve) to the final consumers.

Others that are more consumer-oriented are presented by Sakhare et al. [8], who have coupled price tracking with sentiment analysis to add value to the decision-making process, and by various papers [9], [10], [11], [12], [13], [14], [15] that addressed the topic of real-time scraping systems, comparison engines, and mobile alert applications. These experiments showed that it was possible to create platforms capable of tracking prices between multiple locations, notifying, or visualizing simple trends. Nevertheless, the majority of them were not multi-platform scalable, continuously and automatically automated, easily customizable alerts, or visualization of price-history over an extended time span - features required in a holistic and dependable price-tracking service.

Despite the fact that the contributions make great progress towards the field, most of the available works are limited to single-platform scraping or only solve the dynamic pricing models of companies or do not use continuous monitoring but instead take snapshots at regular intervals. There is hardly any to offer a single system that incorporates multi-platform scraping services, historical trends visualization, user-generated alerts, automatic scheduling, and all these into one consumer-oriented application.

Cost Curve tackles this gap by providing a commodious platform free of charge that allows the user to monitor price fluctuations of a product over time, visualize it, cross-platform trends, and get a personal notification when a product hits the desired threshold. The system aims to offer a more data-heavy online shopping experience by filling the gap between the academic approaches to dynamic pricing and the actual needs of consumers.

3. METHODOLOGY

The approach that will be used in this project is based on a systematic approach towards designing, developing and testing a multi-platform price-tracking solution that will be in a position to get real-time product data on the Indian e-commerce websites. The entire process is categorized into four key steps, namely Data Collection, Data Processing, System Integration and generation of the output.

3.1. Data Collection (Web Scrapping Layer)

The data of products is automatically gathered in several e-commerce websites in India including Amazon, Flipkart, Snapdeal, Naaptol, Shopsy, JioMart, Meesho and IndiaMART. Billions of current web pages were searched with a custom Python scraping engine written using Requests and BeautifulSoup and optionally Selenium WebDriver was used to load any JavaScript-based page.

The scraper uses mobile/desktop user-agent rotation, stochastic delays, URL normalization and HTML selection/extractor to extract the following fields:

- Product title
- Current price
- Product URL
- Image link
- Platform name
- Availability status

All of the platforms are scraped with special scraping functions and high reliability with anti-bot bypass mechanisms like dynamic headers, stealth browser configurations or fallback parsing as well as fallback parsing strategies.

3.2. Data Processing & Cleaning

Raw product data are extracted and processed through various steps to make it consistent:

- Noise filtering: the extraneous extraneous results are eliminated with the help of keyword matching.
- Normalization: price data is changed to integers; currency is changed to INR.
- Fallback processing: the loss of a field initiates regenerates or different extract routes.
- De-duplication: duplicated entries in platforms are integrated.

These measures enhance precision and minimize the false positive in subsequent comparison.

3.3. System Integration & Architecture:

The backend of the system will combine the scraper as a part of a single service that may be manually or periodically evoked. The workflow includes:

- Approving a query for the user (search term).
- Requested all the platform-specific scraping modules concurrently.
- Combining all results obtained to a structured JSON.
- Saving information in the database where price-trend analysis is to be done.
- Firing alert logic when prices are observed to fall.

The scraper is constructed as a reusable component interchangeable with being slotted into a microservice-based structure or effortlessly place like REST APIs.

3.4. Output Generation & User Delivery

The refined and summed data is delivered in JSON format so that it can be used in:

- Web application UI (React frontend)
- Price comparison views
- Historical price tracking graphs
- Notification pipeline for price-drop alerts

The end results of processing are delivered to the users at once, as they obtain them through search, or at the regularly scheduled times, in accordance with planned activities.

3.5. Evaluation Framework

Evaluation parameters also form part of the methodology and include:

- Scraping accuracy across platforms
- Response latency
- Ability to bypass anti-bot restrictions
- Consistency of price extraction
- Successful alert triggering for valid drops

This will guarantee that the system will be tested on the actual constraints of the dynamic e-commerce sites.

4. SYSTEM ARCHITECTURE

Cost Curve system adheres to a modular design that can be easily used to gather, process and display the price of products on various e-commerce sites. The architecture comprises of five major components, these include the user interface layer, the layer of backend service, the web scraping engine, the data processing module, and the database system. These elements are interconnected to provide the ability to track the prices of their products in real time, compare them, and notify the user.

On the user interface level, there is a web-based frontend created with React that gives the user an interactive interface where they can search their products, compare prices across various e-commerce websites, and control their own watchlist. Price trends are also shown on the interface and the user can set prices limits on which notifications can be set.

The service layer of the system is the central control unit that is called the backend service layer. It is implemented on a Python-based API framework, accepting user requests, authenticating and coordinating communication between the frontend and scraping modules and the database. With the search of a product, the search engine activates the scraping engine and gathers the findings of various e-commerce websites.

The web scraping engine will gather the product information of various e-commerce websites in India like Amazon, Flipkart, Naaptol, JioMart, Meesho, and IndiaMART. Individual platforms are scraped with platform-specific modules in the form of libraries like Requests, BeautifulSoup, and Selenium. These modules are able to get major product characteristics such as product title, product price, product URL, product image link and product availability status.

Data is collected and the information is transferred to data processing and normalization module. Some of the tasks it executes include noise filtering, price normalization to INR format, duplicate elimination and structuring of data. The steps will guarantee that the information gathered is consistent and reliable before it is stored or made available to the user.

The processed data is then put into database layer where it stores the information about the product, historical price history and the watchlists of the users. The data stored is useful in the visualization of historical prices and price-drop detection. Once the system realizes that a product price has gone below a set price limit set by the user, the alert system will fire to alert the user.

The modular design of CostCurve makes the system easy to scale by adding new scraping modules or incorporating more sophisticated analytics elements into the system in the future. Fig. 1 shows the architecture and data flow of the proposed system in general.

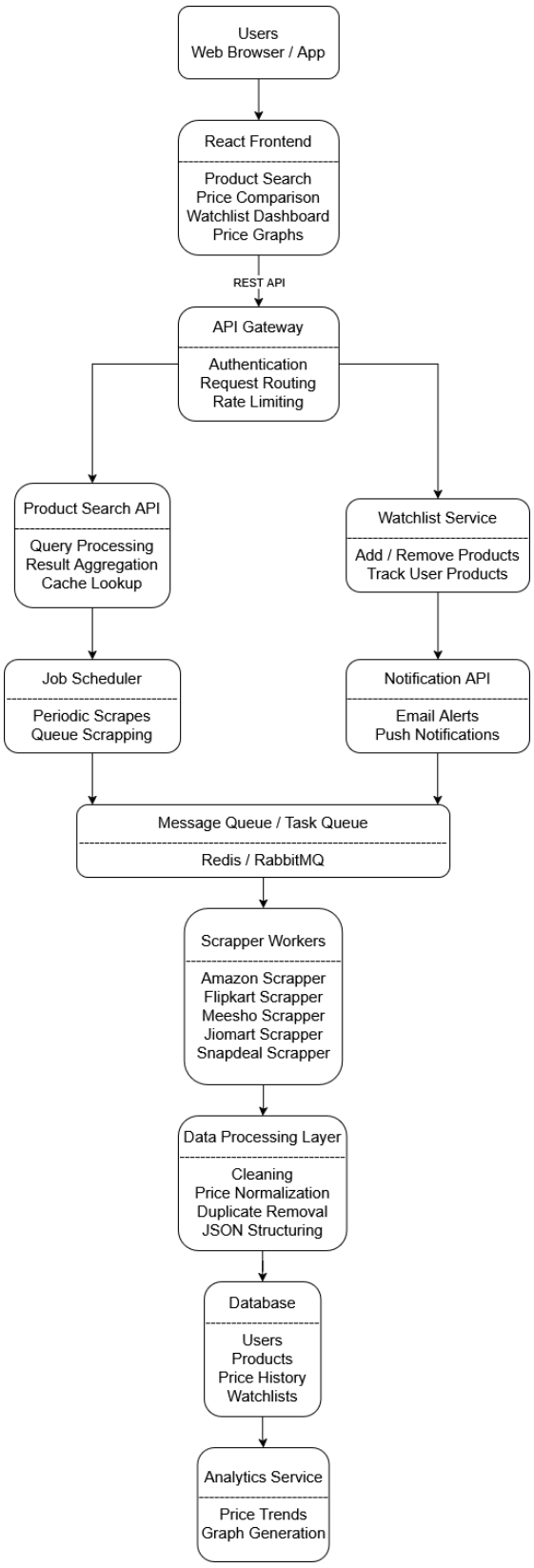

Fig. 1: CostCurve System Architecture

5. RESULTS AND DISCUSSIONS

The creation and incorporation of the CostCurve framework had a few remarkable results that indicated the viability and efficiency of a multi-platform product-tracking and notification system in terms of e-commerce products.

5.1. Functional Implementational Results

- Development of Frontend Interface:

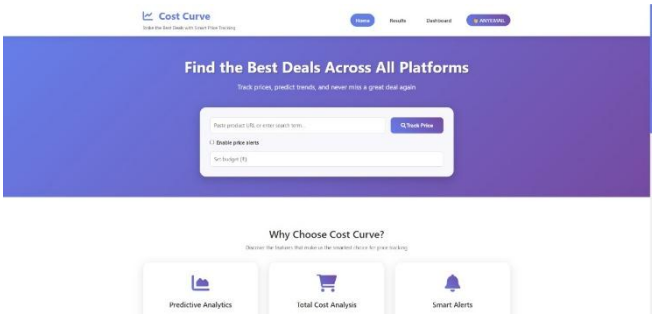

Fig. 2: Frontend of CostCurve

The creation of a fully functioning and responsive front-end to be built in React was developed successfully. The interface is comprised of the home page, product search module, user authentication pages and watchlist dashboard. The design is the focus, it is easy to use, minimalistic, and understandable so that one can find his/her way around the system without any trouble. The interface enables interaction with price-tracking features quickly and the choices being made in terms of the UI/UX prove right after early user testing.

- Web Scrapping on Multiple Platforms

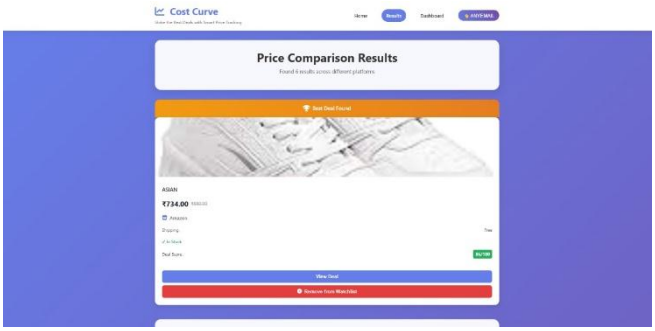

Fig. 3: Page After Searching

It managed to successfully implement the web-scraping engine with Amazon, Flipkart, Naaptol, Shopsy, JioMart, Meesho, and IndiaMART. Product important details such as title, price, URL and image are reclaimed by the scraper with a high degree of reliability.

The findings validate that the scraping functionality can work on a wide range of HTML structure, dynamic element, and platform-specific appearance. The system is capable of creating trusted product comparisons on-the-fly, which shows a good cross-platform compatibility.

- Watchlist and Product Tracking Feature:

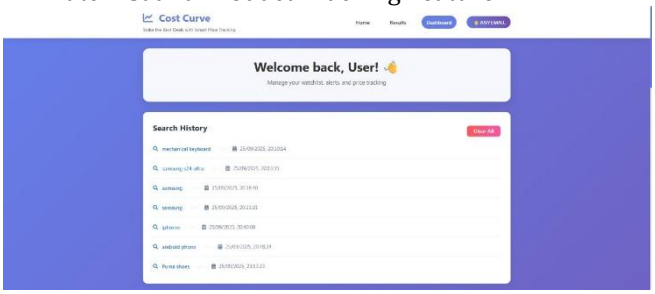

Fig. 4: Watchlist Section

A watchlist module has been implemented into the back end and front end. The users may add a product to their individual watch list where they can access and see the most recent price values of the tracked products.
Persistent tracking between sessions is made possible by this feature and the basis of future mechanisms of alerts. Initial internal errors indicate that watchlist retrieval is precise and up to date as new scraping outcomes have been created.

- **Aggregation of multi-results:**

The system can conduct multi-platform searches with only a single query and compile the results in a common format. The scraper yields structured objects in the form of JSON which can be used to display the results uniformly with all the e-commerce websites.

This demonstrates the practicality of building an unified price-analysis application without using platform API which is usually narrow or uncertified.

5.2. Performance Observations

The test showed that the scraper has a good work performance of moderate loads and provides results in a few seconds, in most cases. Latency was experienced mainly on pages that were highly dynamic (e.g. Amazon), although we reduced these and used fallback parsing approaches and rotating user-agents.

Aggregated results were provided in real time to the frontend with no perceivable bottlenecks in the performance, which proves that the architecture of the system allows providing timely information updates and high interaction rates with the user.

5.3. Discussion

The findings confirm that CostCurve effectively overcomes key limitations that were found in previous literature, including, but not limited to: absence of cross-platform tracking, no trend storage, and the inability to provide user-customizable functionality. The modular scraping engine and watchlist-driven tracking system of the system are a good prerequisite to high-tech extensions, like the automated price-drop notifications, visualization of the historical trends, and predictive analytics.

However, challenges remain. Certain sites can update the HTML layout regularly which also means that scrapers will need to be maintained. Secondly, auto-detection of dynamic rendering and anti-bot techniques may affect the success rate of scraping in the future, so future versions can be enhanced to allow headless browser support or API-based integration when it exists.

All in all, all the applied elements confirm the very idea of Cost Curve, and prove that it can be an effective device to enable consumers to make informed buying choices based on data.

6. FUTURE SCOPE

The existing deployment of Cost Curve creates a powerful purpose of tracking prices on a multi-platform basis, but there are multiple developments that can substantially improve the technology and make it more usable. A potential opportunity involves automated delivery of price-drop notifications, such as email, SMS, or push, as well. With enabling users to specify individual threshold values, the system may send real-time notifications and minimize the necessity of conducting manual re-checks, which allows enhanced user engagement and more usefulness.

The other technological advancement that could occur is the storing of the historical price and representation of the trend. Use of a backend database to document the fluctuation in the prices at all times would enable generation of graphs that would show the overall long-term price trend. The visual analytics could also help the users identify the seasonal trend, approximate the reduction in the future and make superior purchasing decisions. Going into more details, the machine learning systems, such as LSTM or Prophet, can be instantiated to provide predictions about the future fluctuations of a price and may give fore term guidance to buy.

The system too got the chance of being marked to deal with a broader range of e-commerce solutions, which include international market places. This would make sure that Cost Curve would have been implemented on a global background and would strengthen on its competitive knowledge. In addition, scraper robustness might be

improved through the use of browser automation, proxy rotation, and changing adoptions dynamically to enable it to be enhanced in dealing with anti-bot measures that would evolve with time.

On the user experience front future products can acquire extensions to the browser or mobile applications that would allow the user to monitor prices on the product pages. The addition of social functions, such as the sharing of tracked items or community update deals to them, may assist in even better keeping users. Finally, developing an analytics dashboard, which is used to monitor the behavior of the scrapers, performance of the web site, and trend accuracy would assist in maintaining maintainability in the long-term.

Altogether, these would transform Cost Curve into an innovative price-tracking application, a detailed and intelligent price-analysis platform fearing to the future and with real time automation.

7. CONCLUSION

The birth of Cost Curve indicates that one price tracking system can be developed successfully; it can be used to collect the information about products efficiently in different e-commerce systems. The key characteristics that the project had been able to roll out successfully comprise cross-platform search and real-time web scraping and a usable front-end with a working watchlist. All these are compound factors that will debate the key shortcomings that are experienced in the existing literature where most of the systems failed to integrate multi-platform use, centralized monitoring and tailored tracking.

Cost Curve helps to provide the users with unproblematic access to the latest price information by means of the straightforward access to its modular system and the efficient scraping tools and templates create a strong foundation of data-driven decision-making in online purchases. The results disclose that the system applies uniformly across all the major marketplaces and offers a convenient and user friendly prototype to the customers who can agree to the change in the prices of the products.

Though the current form is more concerned with data mining and delivery of basic user experience, the future scope of the work presupposes the saleable model of the future enhancement, which might involve the automated scouts of price drops, the graphic visualization of the historical progressions, and forecast analysis. With these upgrades in place, Cost Curve can evolve itself to become a more comprehensive price intelligence solution that would allow users to make better buying decisions within a more dynamic online marketplace.

REFERENCES

[1] D. Parashar, R. Dixit, A. Dhami, A. Pratap, J. Ali, “Product Price Tracker,” International Journal of Advances in Engineering and Management (IJAEM) Volume 5, Issue 2 Feb. 2023, pp: 01-03

[2] G. T. Cayabyab, R. E. Castillo, P. J. Castro, W. V. Ramos, “Price Watch: An Online Price Monitoring Mobile Application with Exponential Smoothing for Agribusiness and Marketing Assistance Service”

[3] D. Khairkar, S. Dambhare, O. Gupta, V. Thakre, N. Raut, S.N. Dagakar, “E-Commerce Product Price Tracker using Dynamic Pricing Algorithm”

[4] Varun Pilakavil, Hitesh Chaudhari, Pratik Khedekar, Aniket Deotale, Jameer Kotwal, “Real Time Product Price Monitoring & Analysis Application for E-Commerce Websites”

[5] Marcin Nowak * and Marta Pawłowska-Nowak, “Dynamic Pricing Method in the E-Commerce Industry Using Machine Learning.”

[6] Xiaoping Xu, Yuting Wang, T.C.E. Cheng, Tsan-Ming Choi, “Dynamic Pricing Model of E-Commerce Platforms Based on Deep Reinforcement Learning”.

[7] Sentao Miao, Xi Chen, Xiuli Chao, Jiaxi Liu,Yidong Zhang, “Context-Based Dynamic Pricing with Online Clustering”

[8] Nitin Sakhare, Devika Verma, Vikas Kolekar, Avinash Shelke, Akhilesh Dixit, Nikhil Meshram, “E-Commerce Product Price Monitoring and Comparison using Sentiment Analysis.”

[9] M. Anitha, T. Anusha, SK. Fathimabi, P. Durga Sowmya, N. Preethi, “Real-Time Product Price Scraping and Analysis”

[10] D. C. Montgomery, “Cross-Platform Price Comparison System Using Python Scraping (2022)”

[11] Muneeb Iqbal, Faizan Hamayat, Rana Fayyaz Ahmad, Syed Khursheed Husnain, “A Machine Learning-Based Framework for Predicting E-Commerce Product Ratings”

[12] Olamilekan Shobayo, “Innovative Sentiment Analysis and Prediction of Stock Price”

[13] M Premkuma, S. R Ashokkumar, S Anupallavi, Dhamodharan Srinivasa, K Prem, Ajmeera Kiran, “AI based strategic decision framework for precise e-commerce applications”

[14] "Dynamic Web Scraping for E-Commerce Analytics (2022)"

[15] "Mobile Price-Drop Alert App with Cloud Backend (2023)"