

Cost Snipper – Predicting Prices of Online Shopping Items based on Preceding Data



KR. Senthil Murugan , S. Nagajothi, C. Somasundaram, R. Sabareesan, K. Selvakumar

Abstract: *The variation of product prices in online shopping is high which makes it difficult to decide when to buy. The tremendous growth of e-commerce helps us to create the solution of price prediction. We used web scraping technique to get the price data from various online shopping retailers and process the data for each commodity to predict the price for the future which helps us to make decisions on buying online products. We automated the web scraping of data and price prediction daily to make the price available for the customer without any delay.*

Keywords : *Machine Learning, Web Scraping, Variation, Feature extraction, Prediction, Automation, SVM, Regression*

I. INTRODUCTION

With the rapid development of the Internet and data-processing technologies, Web Scraping can be used to explore many possibilities, from Internet news about products or the influence of product price to the influence of sale behavior and important brand strategies. In this paper, we analyze features affecting the price of products, and establish a new model for price prediction. The results show that we have an impact on the sale prices of products, and can improve the accuracy of price prediction over time. Thus, the contribution of this paper is to propose a new prediction model for the price of e-commerce products. Inflation is the tendency of the prices of goods and services to rise in general and continuously [2][3]. High inflation may cause a negative impact on the socio-economic conditions [1].

II. EXISTING SYSTEM

The problem with race-to-the-bottom marketing is that it's short-termism. Retailers think, I need to hit my quarterly KPIs and I'll do anything to make that happen [4]. E-commerce will go from being "reactive" to "proactive": E-commerce

interactions have always been reactive; however, with adoption of machine learning many processes will become proactive

For example: Brands will be able to identify customers who are likely to make a purchase, abandon cart or even return items and personalize their experience. Customers too need to become proactive for enjoying the best experience. We help the customers to overcome the challenges of regular price inflation and variations that commerce sites use to get the high profit.

Moreover, it is not equipped with prediction of price data but only show the data from today and earlier. The system aims for early warning, prediction can be used to know the price of commodities in future, so we can know prediction price data for anticipating. As we use the real data for the prediction system, we can also show the accuracy of our prediction. This helps us to improve the performance and accuracy of the model.

III. PROPOSED SYSTEM

We used web scraping for the price data of products from various retailers. There are a lot of use cases for web scraping: you might want to collect prices from various e-commerce sites for a price comparison site. Or you could even be wanting to build a search engine like Google. Getting started with web scraping is easy, and the process can be broken down into two main parts: acquiring the data using an HTML request library or a headless browser and parsing the data to get the exact information you want.

We used python for the prediction because from development to deployment and maintenance, Python helps developers be productive and confident about the software they're building. Benefits that make Python the best fit for machine learning includes simplicity and consistency [5], access to great libraries and frameworks for AI and machine learning (ML), flexibility, platform independence, and a wide community. These add to the overall popularity of the language.

The best part of programming is the triumph of seeing the machine do something useful. Automate the Boring Stuff with Python frames all of programming as these small triumphs; it makes the boring fun [5]. We automated the web scraping using the CRON scheduling daily to get the price of the product. Cron is the system's main scheduler for running jobs or tasks unattended. A command called crontab allows the user to submit, edit or delete entries to CRON. A crontab file is a user file that holds the scheduling information. You can use crontab to run any of your scripts or one-liners every hour, or maybe just three days a week, it's up to you.

Manuscript received on March 15, 2020.

Revised Manuscript received on March 24, 2020.

Manuscript published on March 30, 2020.

* Correspondence Author

Mr. KR.Senthil Murugan*, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

Ms. S.Nagajothi, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

Mr. C.Somasundaram, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

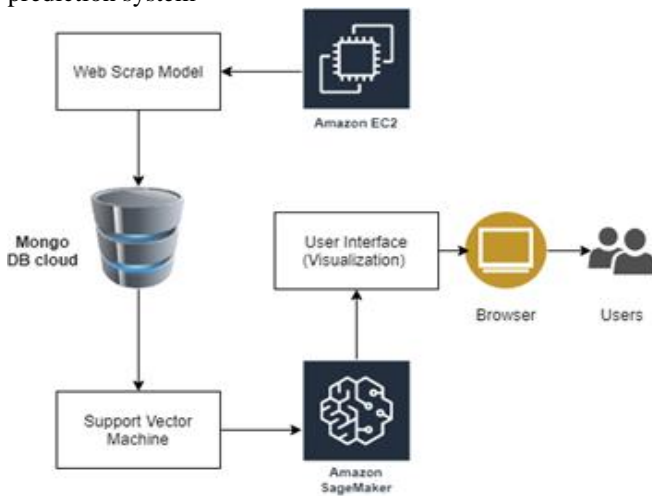
Mr. R.Sabareesan, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

Mr. K.Selvakumar, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Each user can have their own crontab file if they wish, but on large systems root usually disallows this and just uses one main crontab file for the whole system. Root is able to do this because of a file called 'cron.deny' and 'cron.allow' where root can specify who can and cannot have their own crontab [6].

We used the CRON tab to automate web scraping using a node.js script and stored the value in the MongoDB, cloud database. Values from the database are fetched and the predicted prices were calculated with the model for the fast delivery of result to the customers about the predicted price. Thus, in this research we will propose an online price prediction system of consumption commodities. This system helps and supports them in making decisions about their date of purchase and get the profit by effectively utilizing our prediction system



IV. METHODOLOGY

A. Web Scraping

We used node.js for the web scraping. Web scraping is a technique used for retrieving data from websites. You fetch the page's contents, and then extract the data you need from the page for processing, saving it, or simply displaying it on your app. Web scraping mainly involves four steps.

I. Fetching a Webpage

The web scraping first starts with fetching a webpage. The node.js script first accepts the URL as the input and web scrap the details using various node.js libraries. we got the raw HTML from the web page. But now we need to make sense of this giant blob of text. To do that, we'll need to use Chrome DevTools to allow us to easily search through the HTML of a web page. Now, simply click inspect, and Chrome will bring up its DevTools pane, allowing you to easily inspect the page's source HTML. So now we can find the name of the division with this tool for the next step

II. Extracting the data from the webpage

The Chrome DevTools is now showing us the exact pattern we should be looking for in the code (a "big" tag with a hyperlink inside of it). Let's use Cheerio.js to parse the HTML we received earlier to return the name and price of the commodity that is required for the price prediction system. We now ensure that the price and name of the product returned correctly because there may be chance of nested divisions in the division. Now we got

the required data for the system, however a single value cannot be used for training the model.

III. Converting the data to JSON format

The name and price of the product is fetched from the URL. However, it is stored in a single variable which needs to be converted into JSON format for better processing, The JSON format is often used for serializing and transmitting structured data over a network connection.[7]

When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects. This way we can work with the data as JavaScript objects, with no complicated parsing and translations. When storing data, the data has to be a certain format, and regardless of where you choose to store it, text is always one of the legal formats. [7] Here we manually constructed the JSON with the URL, name and price. We now have the required data in a generalized format.

IV. Storing the data into the database

We have the data from the URL but a single value cannot be trained, so we need to save the data daily in the database to get the dataset for the training. We use MongoDB, a non-relational database for storing the data. MongoDB is an open-source and leading NoSQL database. [8] We created a database for storing the price values. Then the connection to the database and collection is made using connection string in node.js. Now the extracted data is inserted into the database. Now the dataset is ready for training the model.

We have the data from the URL but a single value cannot be trained, so we need to save the data daily in the database to get the dataset for the training. We use MongoDB, a non-relational database for storing the data. MongoDB is an open-source document database and leading NoSQL database. [8] We created a database for storing the price values. Then the connection to the database and collection is made using connection string in node.js. Now the extracted data is inserted into the database. Now the dataset is ready for training the model.

V. Automating the web scraping

Now the dataset is ready, however we cannot call the script daily. We need to automate the process so that the script automatically fetches data from the site and store it in the DB. We made the automation using the CRON scheduler. Users that set up and maintain software environments use CRON to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration—though its general-purpose nature makes it useful for things like downloading files from the Internet and downloading email at regular intervals. We used this to schedule our script daily to insert the values of price. The amazon EC2 instance is used because we cannot keep our local machines awake all the time. We pushed the script into Amazon EC2 and scheduled using CRON to run the script daily. Now we have made the required dataset available for the model

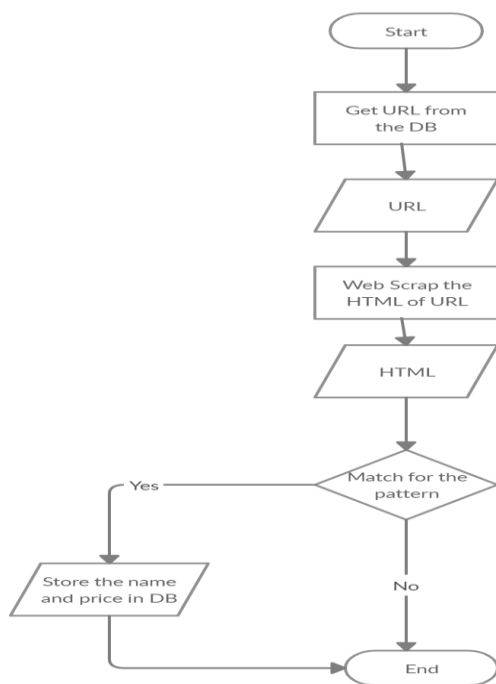


Fig 2. Scraping Process

B. Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

There are 7 major steps involved in the machine learning process. They are data collection, data preparation, choose a model, train a model, evaluate, tuning, predict. We have already completed the data collection and preparation by web scraping.

I. Choose a model

We need to choose the algorithm for the prediction of prices. SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs and predict the value accordingly. We can compare the performances of various algorithms, but SVM gave the best for this price prediction, so based on the input dataset and the performance, we chose the SVM model for the better results and prediction to the customer.

II. Train the model

The main goal of training the model is to quickly predict the value by inputting the dataset. Creating a **train** and test split of your dataset is one method to quickly evaluate the performance of an algorithm on your problem. The

dataset is retrieved from the database and the model for SVM is created and trained with the values of price and date. Here the date is the independent variable and the price is the dependent variable based on the date. The feature we need to predict is the price when we input the date, so the date is converted into the appropriate type and gave to the model as dependent variable and the price as independent variable. Now the model is trained with these values. Now the model is ready for the prediction.

III. Evaluate

Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data. The model is evaluated based on the confidence and the performance of the model. The model is now tested with the data and the performance is improved by repeated evaluation and the confidence improves on further evaluation. Use the precision, recall and accuracy to improve the performance of the model. At the end we will get the best prediction for the model.

IV. Tuning

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data, so the model must be tuned to get the better outcome. Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. The model is tuned by modifying the equations and changing the parameters and by various techniques. Observe the performance and choose the exact parameters and the model is now ready for the best prediction.

V. Predict

Now we have a model that can predict the price for a given date, however to create a reinforced learning model, we automated the training process using the CRON scheduler, so after web scraping the daily price data the model will be trained again to get improved with the new updated dataset and the model will improve day by day, so this automation is scheduled after the web scraping script. Now we have a model that predicts price as well as get improved automatically by the new price variation that occurs. As we have used the real data from the online retailers, we can compare the performance of our model with the original data. Now the model is ready for real-time prediction.

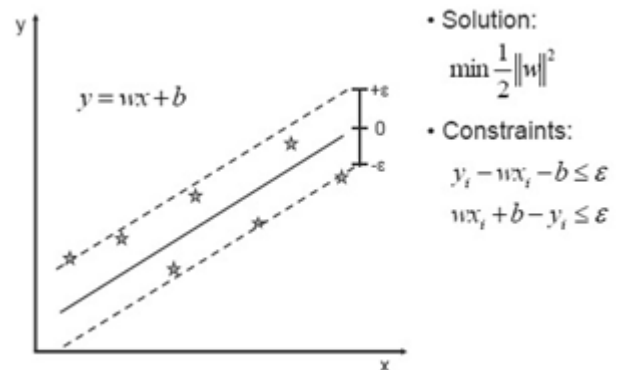


Fig 3. SVM graph and equation

C. User Interface Module

Modern web application based on React.js, to display the predicted price which is stored in the MongoDB. To fetch values from the MongoDB, REST based API to used. From the historical data, graphs will be displayed to visually the price inflation of every product. Graph will also contain the predicted price and actual price. Using this graph, we can visualize the accuracy of the SVM model. Users has to give the URL of the product, application will predict the price and displayed them in user interactive web application. Implementation of the whole system is built using Node.js application. Node.js can be used to build extensive MVC (Model – View – Controller) application, which can be used for banking, e-commerce, and enterprise application.

V. RESULT

The result of this proposed system is to get the exact information from the multiple resource and produce the accurate data to the user by using web scraping and machine learning methodology. In this an online price prediction system of consumption merchandises. This system helps and supports them in making decisions about their date of purchase and get the profit by effectively by the data prediction method.

VI. CONCLUSION

In this research, we have successfully developed an application that will predict the price of online application, using a machine learning algorithm, Support Vector Machine. We compared our proposed system with existing system, the result shows that 82.25% percentage of people believe that, this system will be useful for buying online goods.

REFERENCES

1. “ Introduction of Inflation - BanN Indonesia,” [Online]. Available: <http://www.bi.go.id/id/moneter/inflasi/pengenalannya/Contents/Pentingnya.aspx>. [Accessed: 11-Jan-2020].
2. “ Introduction of Inflation - BanN Indonesia,” 2013.[Online]. Available: www.bi.go.id/id/moneter/inflasi/pengenalannya/Contents/Pentingnya.aspx. [Accessed: 11-Jan-2020].
3. Manual Book of Consumer Price Survey. BPS-Statistics Indonesia, 2013.
4. “Ecommerce trends in 2020: What do the experts predict?” [Online]. Available: <https://econsultancy.com/e-commerce-trends-2020-predictions/>[Accessed: 11-Jan-2020].
5. Seppe vanden Broucke,” Practical Web Scraping for Data Science”
6. [ISBN: 1484235819]

AUTHORS PROFILE



Mr. KR. SENTHIL MURUGAN, M.E., MBA., (Ph.D), (ORCID ID: 0000-0001-6632-0494), Assistant Professor, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India. Membership Details: ¹MISTE, ²IACSIT, ³ICSES, ⁴IAOE, ⁵IS, ⁶IFERP, ⁷IAENG. Email: karusenthilmurugan@gmail.com.



Ms.S.NAGAJOTHI, M.E., Assistant Professor, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India. Membership Details: ¹ICSCS, IAENG, IEEE



Mr.C.SOMASUNDARAM, Student, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India. Email: 16eucs159@skcet.ac.in



Mr. R.SABAREESAN, Student, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India. Email: 16eucs138@skcet.ac.in



Mr.K.SELVAKUMAR, Student, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India. Email: 16eucs150@skcet.ac.in