International Journal of Advances in Applied Sciences (IJAAS)
Vol. 4, No. 4, December 2015, pp. 146~150
ISSN: 2252-8814

Commodity Price Data Analysis Using Web Scraping

M. Kameswara Rao, Rohit Lagisetty, M. S. V. K. Maniraj, K. N. S. Dattu, B. Sneha Ganga
K. L. University, Vaddeswaram, Guntur, Andhrapradesh, India

Article Info

Article history:

Received Sep 20, 2015
Revised Nov 18, 2015
Accepted Nov 29, 2015

Keyword:

Commodity
CPI
Inflation
Scraping
Visualization

ABSTRACT

Today, analysis of data which is available on the web has become more popular, by using such data we are capable to solve many issues. Our project deals with the analysis of commodity price data available on the web. In general, commodity price data analysis is performed to know inflation rate prevailing in the country and also to know cost price index (CPI). Presently in some countries this analysis is done manually by collecting data from different cities, then calculate inflation and CPI using some predefined formulae. To make this entire process automatic we are developing this project. Now a day's most of the customers are depending on online websites for their day to day purchases. This is the reason we are implementing a system to collect the data available in various e-commerce sites for commodity price analysis. Here, we are going to introduce a data scraping technique which enables us to collect data of various products available online and then store it in a database there after we perform analysis on them. By this process we can reduce the burden of collecting data manually by reaching various cities. The system consists of web module which perform analysis and visualization of data available in the database.

Corresponding Author:

M. Kameswara Rao,
K. L. University,
Vaddeswaram, Guntur, Andhrapradesh, India.
Email: kamesh.machiraju@kluniversity.in

1. INTRODUCTION

Web scraping is software technique used for scrapping the data available in various websites and it uses the most universal techniques adopted by many search engines. Web scraping mainly focusses in converting unstructured data available in websites into structured format. The data present in websites is in semi-structured format placed in between html tags. After performing scraping data is stored in local database or spreadsheets to perform analysis.

Commodity price data analysis is performed by collecting the data available in various e-commerce websites. To mine the data available online we are using web scraping. Web scraping is software technique which is used for web data extraction, it indexes the information available on web using a web crawler which is universal technique adopted by many search engines. Scraping can be done by using Python, Java or by using some API's which are available online. In our project we are performing scraping by using SCRAPY tool a python based framework developed for web data extract. SCRAPY contains some predefined libraries which enables us to perform data extract from online and makes our work easier. By scraping the web data we can also perform price comparisons, Weather data monitoring, Website change detection, Research, Web data integration etc. Inflation refers to change in level of prices in an economy over a period of time and CPI measures the price level changes of consumer goods and services purchased by households. Inflation and Cost Price Index (CPI) can be calculated by using data of present and previous years. So, there is a need to maintaining the records of previous years also. Web crawlers goes to a website as specified in the program and collects the data available. After extracting data is maintained in the database to perform various

operations on them. Data available in the database is visualized to the user in the form of bar charts and line charts.

2. RELATED WORK

The availability of on-line costs represents a singular chance for the development of value indexes and also the measure of inflation round the world. As technology is growing day by day the daily need items were also being sold in online market. So to compare the prices of the products that are in online and offline an algorithm is developed which scraps the data such as product name and price and by using this data we can calculate the cost price index (CPI), inflation for particular period of time but disadvantage is that after some time the limitation may recede and it does not include any analysis or number of quantities sold.

Information is largely available on the web data base, if anyone wants to retain some amount of information he must bookmark the web page so to avoid such things search engines are there which searches related information on the entire web. As it is also a type of scraping which extracts the data so we can also scrape the specific data in a website. Some of the search engines are Chickenfoot which is a firefox plugin that provides the programming environment to manipulate or get web page contents and it is written in java script to wrap the content. As it is plugin it was embedded in a web browser and will run slowly as it process the java script and ajax calls. It interacts and scraps the data from the web browser using find() command.

Basically web crawling means a program which goes through the full HTML code of the website by traversing the webpages of the site and gets all the relevant information required for the user is obtained and this is a iterative process for getting more specific information required for the user. The accuracy of the algorithm will be based on the frequency of occurrence of keywords in the webpage and location of keywords in the site.

There are several types of Web Crawling Strategies. They are:

1. Breadth First Search Algorithm
2. Depth First Search Algorithm
3. Page Rank Algorithm
4. Genetic algorithm

2.1. Breadth First Search Algorithm

In this algorithm a uniform search is done along all the neighboring nodes which starts from root node and all the neighboring nodes that are at the same level of root node [1]. If the user required data is obtained then the search is reported as success and the search gets terminated by getting all the required data but if the search doesn't match with the user requirement then it goes down to next level and search will be done at all the neighboring nodes of that level and this process will continue till the user required data is obtained. But when all nodes are searched and the required data is not obtained then it results as a failure.

2.2. Depth First Search Algorithm

In depth first search algorithm, nodes are traversed systematically from the starting of first node and traversing will be done till the end of the last child node. When there are more than one child node then left node will be given the most priority. Then it will be back tracked to all the unvisited nodes till all the nodes are visited [2]. In this algorithm all the nodes are visited once when the breadth is visited [3]. But the disadvantage of this algorithm is that when there are more nodes and branches then it may result a infinite loop [4].

2.3. Page Rank Algorithm

This algorithm helps user in determining the importance of a webpage by calculating total number of citations and backlinks that are present in the webpage [5].

$Pr(W) = (1-d) + d(Pr(l_1)/C(l_1) + ….. + Pr(l_n)/C(l_n))$

Pr(W) --- Page rank for the website that is being calculated

D --- Damping factor of the website

$(l_1 \ldots l_n)$ --- links and citations present in the webpage

By considering human factor a new algorithm of Page ranking is developed by Yougbin Qin and Daoyum Xu [6] and this introduced recommendation mechanism along with page belief and this created balanced rank algorithm of page ranking and gives importance to the needs of users. This effectively avoided topic drift problem.

A new page ranking algorithm with the combination of static algorithm of page rank with classified tree is proposed by Tian Chong [7] where a classified tree is constructed which is used by large number of users and getting similar results while searching and this helps in reducing the problem of theme drift while

searching using only page rank algorithm and outdated pages will be eliminated easily and in turn if increases the effectiveness of the searching and the efficiency of the searching algorithm.

2.4. Genetic Algorithm

Genetic algorithm works using biological evolutions and in it the offspring the fits is obtained by crossing it over the selected best population individuals by using some fitness functions. Some problems present in this but it is suitable for best solution in a specific time [8]. So this suits to the user who has no or time to search a huge database and get very efficient results [9].

2.5. API Tools

Now a day's scraping data from the websites becomes more important and it is also useful for performing analysis on the data collected. We can collect the data using some API tools such as web scraper, import.io, kimono etc., Web Scraper [12] is a tool which is used to create site maps and based on these maps we can navigate through the page for extracting the data. Using selector option in the tool we can navigate and extract different types of data such as links, texts, tables, images etc. Web scraper can also scrape the dynamic data. After extracting we can download the data from the browser as csv.

Kimono [10] is also one of the best API tool for scraping the web data, it is a bookmark in the browser. Kimono provides the functionality of selecting the multiple items such as text, links, and images at a time and can be categorized accordingly in the data modelling option of the kimono bookmark tab. After extracting data it also can be downloaded likewise web scraper tool and it can be further useful in performing analysis. Kimono tool has also the functionality of saving the created API and building apps for the mobile.

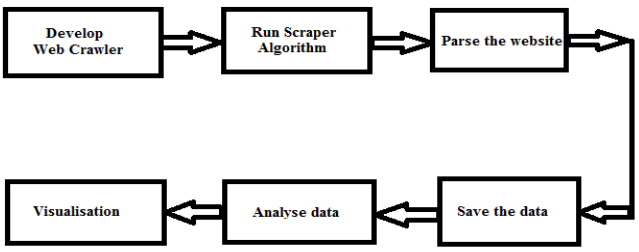
Import.io [11] is another API tool for crawling the data present in the web. It is a browser application which u need to install. Using this tool we can crawl the data within the single page and also the data which is similar in multiple web pages of a website, first we need to select the data what we want and provide column names after that while running the crawler we have to provide the main page url which we had selected the data and also the depth of the pages the crawler needs to navigate so that the crawler navigates accordingly and downloads the data in csv format.

2.6. Beautiful Soup

Beautiful soup [13] is a python library which is used to wrap out the data from the html and xml documents. Using beautiful soup we can scrap the data whatever we want from various websites and it also provides the methods which is implemented in python for navigating, searching and modifying the tree that is parsed which means getting the data whatever we need. To make use of beautiful soup we should install this library in any of the python installers such as pip. For scraping the data first we need to provide the url of the page and then it should be passed to soup method, after that it will parse all the pages and gets the data. The data what we get is along with the tags so to identify and extract the exact data content and get methods are used. Finally the scraped data should be exported to excel file and for this csv library should be imported. Beautiful soup can parse any data whatever we give and fetches different types of data like links, texts etc.

3. PROPOSED METHOD

Data required for price analysis can be collected using searching technique. To collect data using searching technique we initially need to provide keywords of the product to the program then it search for the product name in html code. Once the product name is found the program performs front and backward traversals to find the price of the product and get the price data. But traversing entire html code for product name and price increases the time complexity of the program and the data collected may not be efficient. There may be also some situations where the data is not available in the html code. So, In order to provide efficient method to mine the data we have proposed the following method.



Web crawler perform indexing same as search engine and reaches the website provided by the user. Scraper algorithm helps us to reach to the data and get the data as per the rules provided in the program. We have developed this algorithm using python based framework called SCRAPY. This algorithm parses the information present in the website and save the data into a database and there after we perform analysis on the data using an analysis system. The program used to perform scrapping is called as Spider (user-written classes used for scrapping information). Spider gets the data from the websites based on the domain, xpath and rules provided in the program. Finally a cronjob is written for our program which automatically starts the execution of program and collects data daily as per the time provided by the user.

Our analysis system is a web module developed using html and JSP. It provides an interface for our data to analyse. Once the admin performs login options like add records, modify records, view records and analyse data appears and he can choose any option required by him. This system perform yearly, monthly, daily analysis on the data and makes our work easier to calculate CPI and Inflation. Visualization of data is done using Jfreechart libraries available in java. Using Jfreecharts we can draw graphs for our datasets present in the database. Dataset is passed into the function drawLineChart which plot the graphs and generate the response as an image. Similarly we can also draw column charts and pie charts.

3.1. Algorithm

1. Import packages required for scraping.
2. Create a class which acts as a container for all the objects.
3. Provide details of domain from where data to be scrapped.
4. Provide url's and rules for extraction.
5. Get the xpaths for all the elements to be scrapped.
6. Call parse function.
7. Get connection with database.
8. Pipelining data into database.
9. Run scraper.
10. Define cronjob for running the program in regular interval.

Xpath is provided to the spider in the following way:

Spider is program which crawl into the website to extract data

#This will create a list of products:

buyers = tree.xpath('//div[@title="**Product-name**"]/text()')

#This will create a list of prices

prices = tree.xpath('//span[@class="**Product-price**"]/text()')

4. RESULTS

```
2015-04-06 16:16:04+0530 [scrapy] INFO: Enabled downloader middlewares: HttpAuthMiddleware, DownloadTimeoutMiddleware, UserAgentMiddleware, RetryMiddleware, Defaul
tMiddleware, CookiesMiddleware, ChunkedTransferMiddleware, DownloaderStats
2015-04-06 16:16:04+0530 [scrapy] INFO: Enabled spider middlewares: HttpErrorMiddleware, OffsiteMiddleware, RefererMiddleware, UrlLengthMiddleware, DepthMiddleware
2015-04-06 16:16:04+0530 [scrapy] INFO: Enabled item pipelines:
2015-04-06 16:16:04+0530 [onekirana] INFO: Spider opened
2015-04-06 16:16:04+0530 [onekirana] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2015-04-06 16:16:04+0530 [scrapy] DEBUG: Telnet console listening on 127.0.0.1:6023
2015-04-06 16:16:04+0530 [scrapy] DEBUG: Web service listening on 127.0.0.1:6080
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Arhar-Daal-1Kg/3> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Aashirvaad-Atta-5Kg/130> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Sella-Rice-Loose-1kg/138> (referer: None) ['partial']
C:\products\groceries\groceries\spiders\Spider_1(onekirana).py:40: ScrapyDeprecationWarning: scrapy.selector.HtmlXPathSelector is deprecated, instantiate scrapy.Se
  selector = HtmlXPathSelector(response)
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Tata-I-Shakti-Toor-Dal/989> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Tata-I-Shakti-Moong-Dal/987> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Tata-I-Shakti-Masoor-Dal/990> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Tata-I-Shakti-Chana-Dal/988> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Sugar-1Kg/1271> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Dhara-Refined-Groundnut-Oil-1LTR/1710> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Jaggery-Gur-500gm/2799> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Nature-Fresh-Mustard-Oil-1-Ltr/699> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/TATA-TEA-Premium-1-Kg/75> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Baba-Ramdev-Patanjali-Cow-Ghee-1-Ltr/998> (referer: None) ['partial
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Fortune-Soya-Oil-1-ltr/21> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Fortune-Sunlite-refined-Sunflower-Oil-1ltr/370> (referer: None) ['p
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Amul-Taaza-Homogenised-Toned-Milk-1Ltr/1873> (referer: None) ['part
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Potato-Sugar-Free-1-Kg/2526> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/TATA-Salt-1kg/894> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Onion-Red-1-Kg/2528> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] DEBUG: Crawled (200) <GET http://www.onekirana.com/product/Tomato-Desi-Tanatar-500-gm/2529> (referer: None) ['partial']
2015-04-06 16:16:05+0530 [onekirana] INFO: Closing spider (finished)
2015-04-06 16:16:05+0530 [onekirana] INFO: Dumping Scrapy stats:
        {'downloader/request_bytes': 5030,
         'downloader/request_count': 20,
         'downloader/request_method_count/GET': 20,
         'downloader/response_bytes': 48000,
         'downloader/response_count': 20,
         'downloader/response_status_count/200': 20,
         'finish_reason': 'finished',
         'finish_time': datetime.datetime(2015, 4, 6, 10, 46, 5, 507000),
         'log_count/DEBUG': 22,
         'log_count/INFO': 7,
         'response_received_count': 20,
         'scheduler/dequeued': 20,
         'scheduler/dequeued/memory': 20,
         'scheduler/enqueued': 20,
         'scheduler/enqueued/memory': 20,
         'start_time': datetime.datetime(2015, 4, 6, 10, 46, 4, 836000)}
2015-04-06 16:16:05+0530 [onekirana] INFO: Spider closed (finished)
C:\products\groceries\groceries\spiders\Spider_1(onekirana).py:7: ScrapyDeprecationWarning: groceries.spiders.Spider_1(onekirana).spider inherits from deprecated c
ning only on first subclass, there may be others)
  class spider(BaseSpider):
2015-04-06 16:16:12+0530 [scrapy] INFO: Scrapy 0.24.4 started (bot: groceries)
2015-04-06 16:16:12+0530 [scrapy] INFO: Optional features available: ssl, http11
2015-04-06 16:16:12+0530 [scrapy] INFO: Overridden settings: {'NEWSPIDER_MODULE': 'groceries.spiders', 'FEED_FORMAT': 'csv', 'SPIDER_MODULES': ['groceries.spiders'
```

Figure 1. Scrapped data in Scrapy Framework

Figure 2. Interface for Price Analysis System

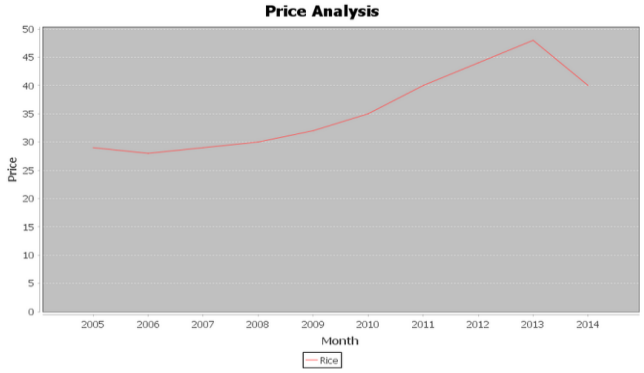


Figure 3. Visualization of scrapped data

5. CONCLUSION

In this paper we have proposed a most efficient technique to mine the commodity prices data from websites and perform analysis on them. This technique resolves all the problems proposed by previous authors on web scraping. We successfully implemented program for scrapping data from online stores and designed a system which perform analysis on the data. The system provides the visualization of data in the form of line charts and also keep track of Inflation and Cost Price Index (CPI) of the country.

REFERENCES

[1] Steven S. Skiena, "The Algorithm design Manual", Second Edition, Springer Verlag London Limited, pp. 162, 2008.
[2] Alexander Shen, "Algorithms and Programming: Problems and solutions" Second edition, Springer, pp. 135, 2010.
[3] Narasingh Deo, "Graph theory with applications to engineering and computer science", *HI*, pp. 301, 2004.
[4] Ben Coppin, "Artificial Intelligence illuminated", Jones and Barlett Publishers, pp. 77, 2004.
[5] Sergey Brin and Lawrence Page, "*Anatomy of a Large scale Hypertextual Web Search Engine*", Proc. WWW conference, 2004.
[6] Yongbin Qin and Daoyun Xu, "A Balanced Rank Algorithm Based on PageRank and Page Belief recommendation".
[7] TIAN Chong, "*A Kind of Algorithm For Page Ranking Based on Classified Tree In Search Engine*", Proc International Conference on Computer Application and System Modeling (ICCASM 2010), 2010.
[8] S. N. Sivanandam, S. N. Deepa, "Introduction to Genetic Algorithms", Springer, pp. 20, 2008.
[9] S. N. Palod, Dr. S. K. Shrivastav, Dr. P. K. Purohit, "Review of Genetic Algorithm based face recognition", *International Journal of Engineering Science and Technology (IJEST)*, Vol. 3, No. 2, Feb 2011.
[10] https://www.kimonolabs.com/
[11] https://import.io/
[12] http://webscraper.io/a
[13] Vineeth G. Nair, "Getting Started with Beautiful Soup", Packt Publishing Ltd.