Product Cataloging And Intelligence

Team No: 30

Shailendra Kumar Joshi

Karan Mangla

Karan Agarwal

Overview

Our job was to extract product data from our target website (amazon.com) and enriched our master dump from which we extracted the product dump and vendor dump. After getting required details about each product including their specifications we performed analytics over this data on the basis of price and ratings to get the best price and vendor details out of them.

Goals

1. Generation and Enrichment of master dump from target website.
2. Extraction of product and vendor dump from master dump.
3. Escaping target website protocols and bots in order to access the product details.
4. Analytics over vendor and product data to get the best one out of them.However several results may be possible in case of same price and ratings.

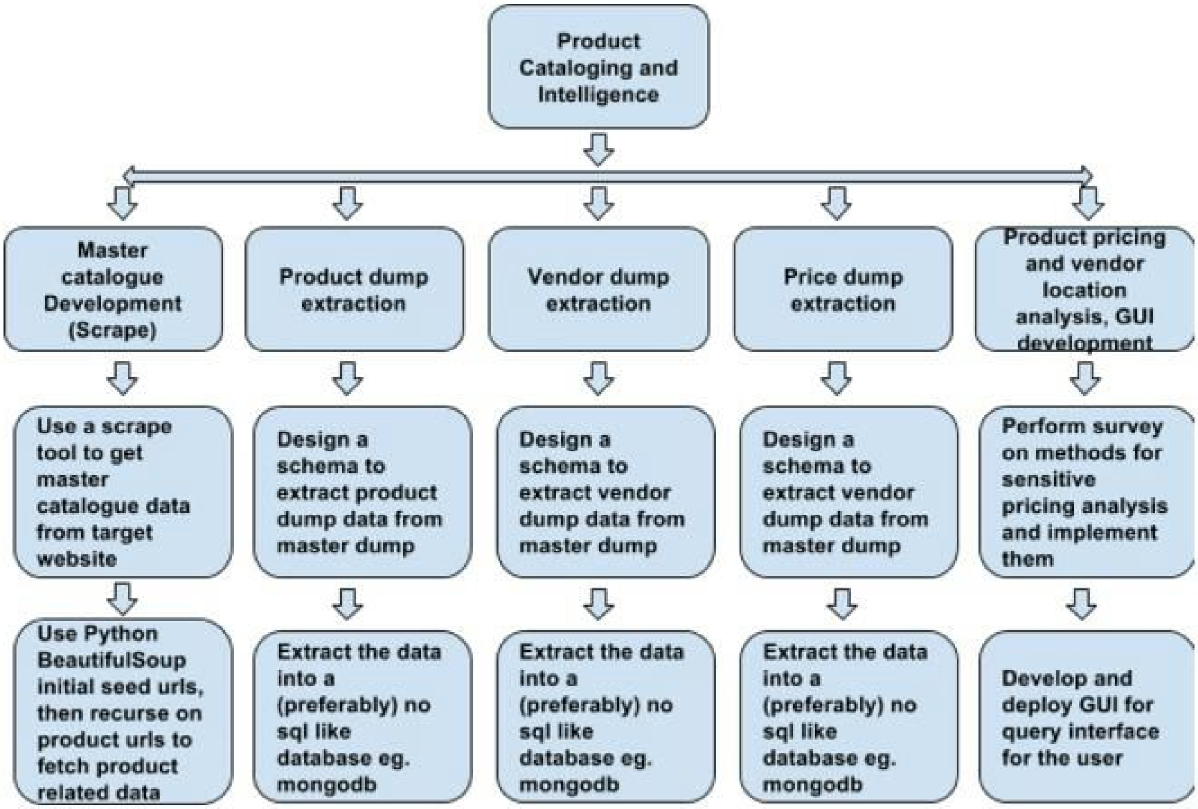
Dataset

We created our master dump from target website and updated it periodically in order to get the latest details regarding product, price and vendor update changes.

Our Work

- Generation and enrichment of master dump from target website.
- Generation and enrichment of vendor and product dump from that master dump
- Updation of master dump periodically in order to update details
- Escaping from site bots and protocols
- Analytics over the product and vendor data in order the get the best product on the basis of analytical results.

Workflow Followed


Product Cataloging and Intelligence
Master catalogue Development (Scrape)
Product dump extraction
Vendor dump extraction
Price dump extraction
Product pricing and vendor location analysis, GUI development
Use a scrape tool to get master catalogue data from target website
Design a schema to extract product dump data from master dump
Design a schema to extract vendor dump data from master dump
Design a schema to extract vendor dump data from master dump
Perform survey on methods for sensitive pricing analysis and implement them
Use Python BeautifulSoup initial seed urls, then recurse on product urls to fetch product related data
Extract the data into a (preferably) no sql like database eg. mongodb
Extract the data into a (preferably) no sql like database eg. mongodb
Extract the data into a (preferably) no sql like database eg. mongodb
Develop and deploy GUI for query interface for the user

Master catalogue Development:

- Scrapping of data from target website to generate master dump.
- Enrichment of master dump periodically in order to get updated details.
- Escaping from website's protocol which can block our access to that site.

Product Dump Extraction:

- Generation of product dump from master dump.
- Enrichment of product dump whenever master dump updated.
- Extraction of details from product dump which includes variable number of fields for different products
- Saving of product details into a schemaless database(MongoDB)

Vendor Dump Extraction:

- Generation of vendor dump from master dump.
- Enrichment of vendor dump whenever master dump updated.
- Extraction of details of vendor including their offered price.

Analytics Over Data:

- Perform price based analytics of data that results best price in Market. Results may be multiple in case of same price)
- Perform ratings based analytics of data that results vendors that have good reputation in market. (Results may be multiple in case of same ratings)

Pipeline Followed for Scraping data

A general framework was created which would scrape out the data given any URL link on the amazon website. Hence all that was needed was to first crawl off the links pointing to the products and then the generalised crawler would scrape off all the relevant information about the products needed. As a sitemap.xml is not available for amazon.on, we had to scrape off product links corresponding to products belonging to various kinds of categories like mobiles, laptops, headphones and the like. Hence a list of seed URLs was scraped for each of the categories and then fed into the generalised crawler which would be enough standalone in order to get the required details.

Tools Used

1. Python

   Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.The language provides constructs intended to enable clear programs on both a small and large scale.

   Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

We used python for most of the text processing and db access that was required by us.

2. BeautifulSoup

Beautiful Soup parses a (possibly invalid) XML or HTML document into a tree representation. It provides methods and Pythonic idioms that make it easy to navigate, search, and modify the tree.

A well-formed XML/HTML document yields a well-formed data structure. An ill-formed XML/HTML document yields a correspondingly ill-formed data structure. If your document is only locally well-formed, you can use this library to find and process the well-formed part of it.

This is used by us primarily for parsing the retrieved web pages.

3. Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed. Flask is called a micro framework because it does not presume or force a developer to use a particular tool or library. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

This was used by us primarily by us for handling the search portal where the user can put in his queries and get a vendor level analysis about the product that he has searched upon.

Challenges Faced

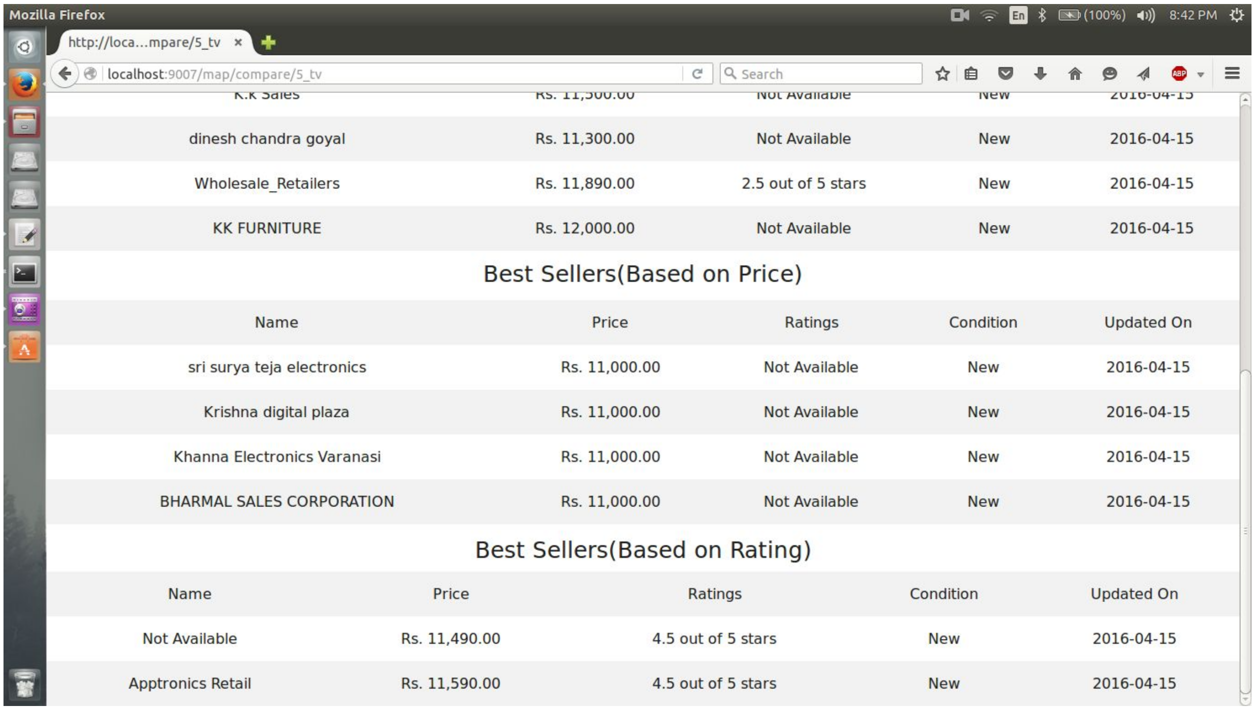
- Intelligence to avoid screen scraping blockages.
- Writing a generalised scraper that would work site along across all pages on a specific website.
- Use of MongoDB due to big data of the target website.
- Use of Flask as a framework due to its specific properties.

(No database abstraction layer and no form validation)

Output

We got result on the basis of analytics which includes vendor's price that are best in market and vendors whose ratings are quite good.

Here is an example screenshot of our analysis.



K.K Sales	Rs. 11,500.00	Not Available	New	2016-04-15
dinesh chandra goyal	Rs. 11,300.00	Not Available	New	2016-04-15
Wholesale_Retailers	Rs. 11,890.00	2.5 out of 5 stars	New	2016-04-15
KK FURNITURE	Rs. 12,000.00	Not Available	New	2016-04-15

Best Sellers(Based on Price)

Name	Price	Ratings	Condition	Updated On
sri surya teja electronics	Rs. 11,000.00	Not Available	New	2016-04-15
Krishna digital plaza	Rs. 11,000.00	Not Available	New	2016-04-15
Khanna Electronics Varanasi	Rs. 11,000.00	Not Available	New	2016-04-15
BHARMAL SALES CORPORATION	Rs. 11,000.00	Not Available	New	2016-04-15

Best Sellers(Based on Rating)

Name	Price	Ratings	Condition	Updated On
Not Available	Rs. 11,490.00	4.5 out of 5 stars	New	2016-04-15
Apptronics Retail	Rs. 11,590.00	4.5 out of 5 stars	New	2016-04-15