"RANKING ACADEMIC INSTITUTIONS"

Project Report Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

Shreya Srivastava (141224)

Khushboo Bansal (141329)

Under the supervision of

Dr. Suman Saha

Assistant Professor (Senior Grade)

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

May 2018

CANDIDATE’S DECLARATION

We hereby declare that the work presented in this report entitled **“Ranking Academic Institutions”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2017 to May 2018 under the supervision of **Dr. Suman Saha,** Assisstant Professor, Dept of CSE, JUIT**.** The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Shreya Srivastava, 141224 Khushboo Bansal, 141329

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Dr. Suman Saha,
Assisstant Professor,
Department of Computer Science & Engineering and Information Technology (CSE&IT)

Dated:

ACKNOWLEDGEMENT

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide Dr. Suman Saha, Department of Computer Science and Information Technology at Jaypee University of Information Technology, Solan in bringing this report to a successful completion. We are grateful to Prof. Dr. S.P. Ghrera, Head of Department of Computer Science and Information Technology for permitting us to make use of the facilities available in the department to carry out the project successfully. The project is helping us in expanding our knowledge in the domain of data sciences, text analysis and big data.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

With the ever increasing need to gain information specific to our queries, from the humungous amounts of data available on the internet today, technologies like web crawler, text mining and big data are indispensable. **College and university rankings** are rankings of educational organizations and institutes in higher education that have been ranked on the basis of miscellaneous conjunctions of sundry factors. In addition to ranking the entire institutions, organizations carry out rankings of peculiar programs, departments, and schools A web crawler (also known as a web spider or web robot) is a program or automated script which browses the World Wide Web in a methodical, automated manner. This process is called Web crawling or spidering. Many legitimate sites, in particular search engines, use spidering as a means of providing up-to-date data.

Companies use text analysis to set the stage for data-driven approach towards managing content. The moment textual sources are sliced into easy-to-automate data pieces, a whole new set of opportunities opens for processes like decision making, product development, marketing optimization, business intelligence and more.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. The importance of big data doesn’t revolve around how much data you have, but what you do with it. You can take data from any source and analyze it to find answers that enable 1) cost reductions, 2) time reductions, 3) new product development and optimized offerings, and 4) smart decision making.

The current systems used for ranking the universities, while useful, in our view, does not sufficiently capture the vital innovative spirit of higher education institutions. Is entrepreneurship promoted among students? Is an incubator or accelerator hosted on campus? Are students being given opportunities, as part of their studies, to pursue internships or training in industry.

CHAPTER 1

INTRODUCTION

1.1 Introduction

College and university rankings are rankings of educational organizations and institutes in higher education that have been ranked on the basis of miscellaneous conjunctions of sundry factors. In addition to ranking the entire institutions, organizations carry out rankings of peculiar programs, departments, and schools. There is a hunger in public, among policy makers and the audience we deal with most among institutions amongst colleges themselves to try to figure out what values they are providing to students. College costs are going up dramatically. They have been going up faster and faster than the rate of inflammation and students are increasingly struggling to afford the prices that are being charged by colleges. Students, policymakers and colleges, everyone wants to know that with those rising costs what kind of bang students are getting for their colleges buck and that is what's really driving the rankings. The rankings all have a different focuses. Some of them like US news and World Report and some others typically look at labor market outcomes for students, average SAT scores that students enter with, the quality if the students and faculty that are coming in meaning how well prepared they are. The challenge with those kind of rankings is that they do not typically give colleges rewards for equity for keeping their doors as open as possible nor do they look at how much students grow while they are in college.

There are crucial dissimilarities in the ranking methodologies that are used. These distinction can be seen in the definitions of what constitutes quality, in the gauge and scales used to measure quality. These distinctness result in variety of ranking approaches. Accordingly, the ranking results differ substantially from one ranking approach to another.

Spyder IDE

Spyder is a powerful interactive development open source cross-platform IDE for scientific programming with advanced editing and testing in the Python language. As compared to other IDEs, Spyder has a unique set of features- cross platform, open source written in python and available under non-copy left license. Features of Spyder are, it provides user an editor with syntax highlighting and introspection for code completion, provides support for multiple Python console and also the ability to explore and edit variables from a GUI Plugins.

Anaconda

Anaconda is bunch of Python packages plus a package manager called Conda. These packages are very popular in Data Science communities. Some of these packages are NumPy, SciPy, etc. If we install anaconda, we do not need to install packages individually. We get all of these in one shot.

Python

Python is a simple, general purpose and very powerful high level programming language. It supports multiple programming paradigms including interactive, object-oriented, imperactive and also has a comprehensive and large Standard library. It is simple for server automation. It has a huge library for building web apps. The language itself is enough for handling HTTP and is also heavily used for scientific computing for example, earthPy, for earth sciences, AstroPy for astronomy and others.

Web Crawler

How is it that google is able to bring the exact search results when you type in a query given the fact that there are trillion of pages on the internet? Behind the scene, it is the web crawlers that are at work.

Web crawler is an automated script which browses through the internet in an organized manner. The web crawler examines the keywords in the pages, the kind of content each page has and the links, and returns the information obtained to the search engine. A web crawler collects pages from the web and then, indexes them in a methodical and automated manner to handle search engine queries. When a search query is entered, these crawlers scan all the relevant pages that contain these words and turn it into a huge index.

Crawlers also help in validating HTML codes and checking links.

The spider starts its crawl by going through the websites or list of websites that it visited the previous time. When the crawlers visit a website, they search for other pages that are worth visiting. Web crawlers can link to new sites, note changes to existing sites and mark dead links.

Talking about the methodology used in Google inside search, crawling is done across over a staggering 60 trillion pages and relevant results are brought back. This is done by indexing. Site owners have the right to decide which pages they want to get indexed. Indexing is done by sorting and looking at the quality of content and other factors. Google then with the help of other features like spell checking, auto complete, search methods other than text like voice and image recognition, understanding synonyms etc provide a better search experience.

After the web crawler completes collecting all the data from various sources, the data remains in an unstructured form, mainly in JSON, CSV or XML formats. This is raw data and gleaning useful insights from it, is known as data mining. The seriousness and importance of data mining comes to light during the extraction process because you've got to deal with web pages errors, data in multiple languages and irregular markups. Retaining the encoding format in its original form is also important.

Web crawling and Data mining cannot be completed without Data extraction. Data extraction is extremely useful for people indulging in online shopping. There are sites with data sources that are structured, Amazon for instance, but some remain unstructured and are hidden deep in the web.

To get the data from such sites, the query will have to be entered in the search box and filters are narrowed to get the results. The search query is answered in the form of product details embedded in HTML.

Only a special crawler that can parse HTML can scrape and extract exact product details as demanded by the user. The details may include product title and information, pricing, variations, rating, reviews, product code and so on. The feed is updated regularly, so the user gets only relevant and up to date data.

Uses of Web Crawler:

Web crawlers have become so important to companies having a strong online presence, and they use it to obtain data like product information, reviews, pricing details and images to ensure they deliver better than their rivals. Web crawlers can, thus, make an impact on every aspect of business.The presence of web crawlers makes all the difference to the end user. Everywhere businesses are looking for ways to beat their competition trying to provide better quality products at reasonable prices.

<u>Real Estate:</u>

This catalog is prepared by noting the property descriptions as per to type, number of bedrooms, images, market value and other relevant information in a structured format.

The buyer/seller can visit the website offering such information and browse through the listings to know the price and other details of a particular property.

<u>Automobile Industry:</u>

Data that is required by the clients are vast in amount and it is to be explored from enormous resources like auto spare parts sites, automobile communities, blogs and the like.

The web crawler goes through all the source sites provided by the client, collects and extracts the required data. It is also important to set the parameters for data extraction separately for each site since the source websites may have different structure and design. The user can compare the prices, observe the latest trends, and other data delivered by different sources and then make wise decisions.

Text Analysis

Text analysis is about parsing texts in order to extract machine-readable facts from them. The purpose of text analysis is to create sets of structured data out of heaps of unstructured, heterogeneous documents.

The process can be thought of as slicing and dicing documents into easy-to-manage and integrate data pieces. Text analysis helps translate a text in the language of data. And it is when text analysis prepares the content, that text analytics kicks in to help make sense of these data.

Companies use text analysis to set the stage for data-driven approach towards managing content. The moment textual sources are sliced into easy-to-automate data pieces, a whole new range of opportunities opens for processes like decision making, product development, marketing optimization, business intelligence and more. When turned into data, textual sources can be further used for obtaining valuable information, discovering patterns, automatically managing, using and reusing content, searching beyond keywords and more.

Big data

Hadoop is an open-source software framework that is itself written in Java, is used for storing data and running applications on clusters of commodity hardware, single or multiple. It provides massive storage for any type and variety of data, providing extensive and expansive processing power and also the ability to handle illimitable and concurrent tasks or jobs virtually.

Importance of Hadoop

- **Ability to store and process huge amounts of any kind of data, quickly.** With huge volume of data and its varieties constantly increasing at a high velocity, especially from the sources like social media, IoT, etc. is the key consideration.
- **Computing power.** The distributed computing model processes big data faster than the traditional ones. The more number of computing nodes you use, the more processing power you have.
- **Fault tolerance.** Data and application processing are protected against hardware failure as the hardware used is low-cost maintenance. If a single node goes down or fails to perform it's task, jobs are automatically rechanneled to other nodes making sure that the distributed computing does not fail altogether. Multiple copies of all data are created and stored automatically.
- **Flexibility.** Unlike traditional relational databases, user does not have to preprocess the data before storing it. User can store as much data in volume as he wants and can decide how to use it later. The huge volume of data includes unstructured data like text, images and videos.
- **Low cost.** The open-source framework is free of cost and it uses commodity hardware to store large and complex quantities of data.
- **Scalability.** One can easily grow the system by simply adding more computing nodes to handle the data. Little administration is required.

1.2 Problem Statement

Rankings are used by some government departments in their higher education policy, by institutions looking for international partners to collaborate on various grounds such as innovative projects, student exchange programs etc. and by prospective students searching for a place to study.

The current systems used for ranking the universities, while useful, in our view, does not sufficiently capture the vital innovative spirit of higher education institutions. Is entrepreneurship promoted among students? Is an incubator or accelerator hosted on campus? Are students being given opportunities, as part of their studies, to pursue internships or training in industry? Moreover is the institute helping the students in overall development by promoting sports? And if the institution is providing all such facilities, then what are the actual figures of their success.

1.3 **Objective**

The main objective of this project is to study the concept of web crawlers, text analysis and big data and implement these three to obtain a unique ranking of universities based on non-conventional criteria such as incubation and innovation opportunities and sports infrastructure. The major task would be to decide upon the parameters and build an algorithm upon which the universities would be ranked.

The motive is to become completely aware and familiar with the technology used for the implementation of the project and make the best use of it for our project completion. Also we hope that through this project we are able to provide some useful results for the concerned persons or organizations who might be interested in knowing these details.

1.4 Methodology

1. **Procuring the data:**

   The foremost step is collecting the data required for ranking the universities on the basis of three parameters:

   a. Student to faculty ratio
   b. Enrollment of foreign students
   c. Scholarships awarded by a university

   This step is accomplished by deploying web crawler on publicly accessible data provided by the government website, keeping in mind the above mentioned parameters. The links related to the desired information are extracted and the final data is obtained in the form of a csv extension file.

2. **Processing the data:**

   The data obtained in three separate excel files corresponding to the parameters discussed above is processed and transformed into a single file for applying the ranking algorithm. The individual datasets are cleaned in order to glean only the relevant information. The cleaning process involves removing unnecessary white spaces and punctuation marks, sorting the datasets as per names of the colleges, discarding the unwanted columns, summing the entries in the remaining columns to get concrete data. Finally an algorithm is implemented using Python scripting language which takes all the three cleaned datasets as input, extracting rows having college names common in all the datasets in all the three datasets and produce a single file.

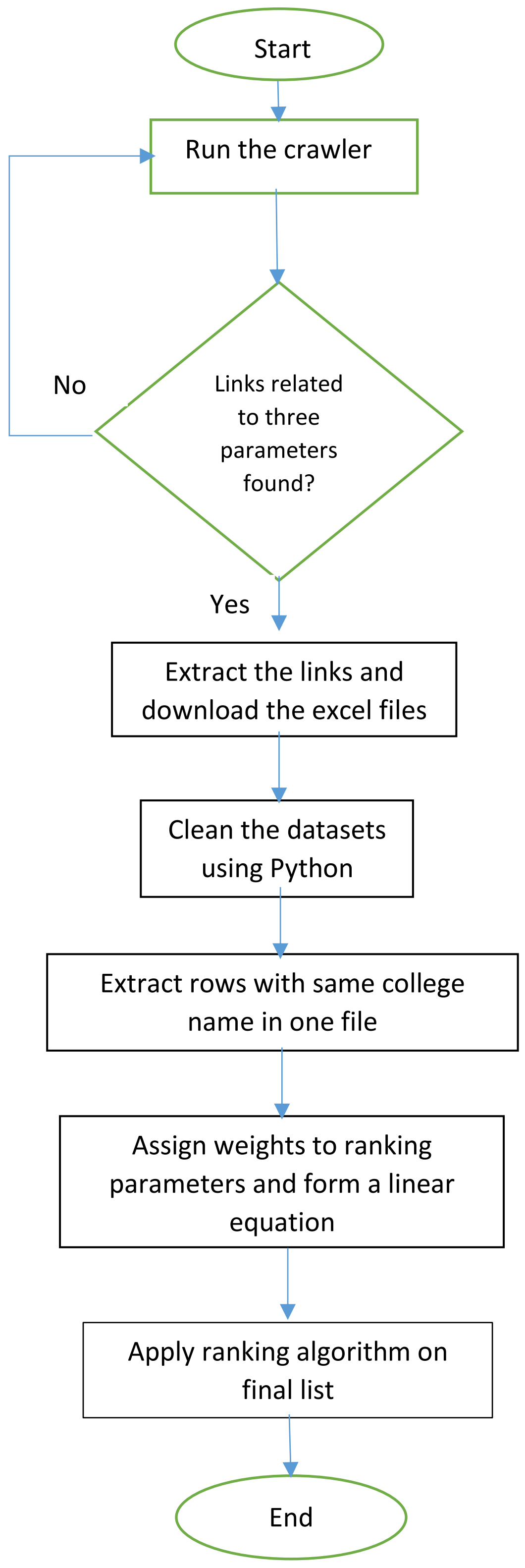
3. **Ranking Algorithm:**

   A ranking algorithm is applied to the final dataset. First, on the basis of our research on the internet, appropriate weights are assigned to the three ranking parameters and a value is calculated corresponding to each college entry using a linear equation with three variables and their coefficients as:

$$\mathbf{Y = ax_1 + bx_2 + cx_3 + const.}$$

This value is stored in a separate column corresponding to all the entries.
A simple sorting algorithm is then applied using this calculated value and hence, resulting in the list of universities that are ranked in ascending order.

Flowchart:


Start
Run the crawler
Links related to three parameters found?
No
Yes
Extract the links and download the excel files
Clean the datasets using Python
Extract rows with same college name in one file
Assign weights to ranking parameters and form a linear equation
Apply ranking algorithm on final list
End

LITERATURE SURVEY

2.1 Title: PyBot: An Algorithm for Web Crawling [AGK Leng, Ravi Kumar P]

The authors have thoroughly talk about the functioning of web crawlers, highlighting the steps which a standard search engine takes to provide results to a query which include, crawling, indexing and searching[1]. The authors also discuss the rules that crawlers should follow so as to avoid getting banned from the website or/and result in unfavorable network conditions. The paper elucidates various types of crawlers proposed and implemented over the years like general purpose crawlers, focused crawler and distributed crawler. Also five different kinds of search engines have been discussed. The authors have laid down certain criteria based on which the outcome of a web crawler can be evaluated which are: Selection policy, Re-visit policy, Politeness Policy and Parallelization policy. The paper also mentions some commercially functional search engines with pertinent details like FAST Crawler, RBSE, WebFountain etc. Finally, the authors have proposed their own crawler, PyBot using Breadth First Search. This search takes in consideration the methodology in which first, the root node is expanded and following that, all the successors of the root node are expanded, then their successors, and so on.

2.1.1 Introduction

The three important sequential tasks that a standard search engine does as shown in Figure 2.1.1 are:

- Crawling
- Indexing
- Searching

Crawling is the most essential one among these. The prime purpose of a crawler is to download web pages for indexing, and other purposes like page validation, structural analysis , visualization, update notification, mirroring and for spam purposes like harvesting email addresses etc.

The indexing module provides information about ranking of pages back to crawlers so that the crawlers can collect the important pages first as shown in Figure 2.1.1.

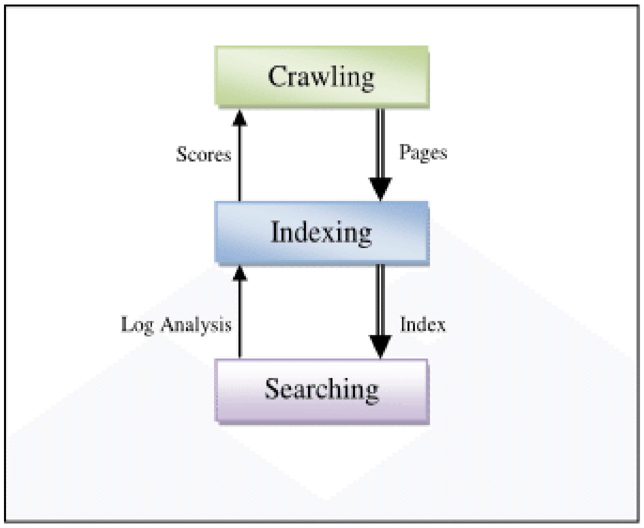


Fig 2.1.1 Ranking of pages

Some of the crucial jobs performed by a crawler are downloading a web page, parsing through the downloaded pages and retrieving all the hyperlinks and for each link retrieved , repeating the process. Once a URL is specified , the crawler follows all he links found in that page and visit all those links and download the pages. It continues until there is no link left to be explored or the crawler wants to halt. A crawler treats a Web site as a tree-structure as shown in Figure 2.1.2. Here the start URL is the root of the tree and the children of the root are their links. Some children may not have any links like the child 2 of root and they are called as dangling child.

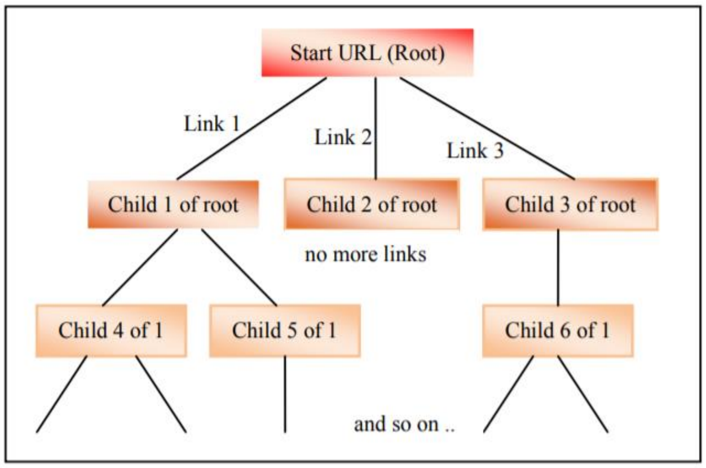


Fig 2.1.2 Website as tree structure

2.1.2 Type of crawling:

General Purpose Crawling:

A general purpose Web Crawler gathers as many pages as it can from a particular set of URLs and their links.[2] In this, the crawler is able to fetch a large number of pages from different locations. But, general purpose crawling can slow down the speed and network bandwidth because it is fetching all the pages.

Focused Purpose Crawling:

The aim of the focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. It only crawls the desired regions of the web and leads to significant savings in terms of both hardware and network resources. The most important evaluation of focused crawling is to measure the harvest ratio, which is the rate at which relevant pages are acquired and irrelevant pages are efficiently sieved from the crawl. This harvest ratio must be high, or else the focused crawler would be spending a lot of time in solely eliminating the irrelevant pages, and it would then be better to use an ordinary crawler instead of the focused crawler.

Distributed Crawling:

In distributed crawling, multiple processes are employed to crawl and download pages from the Web thus, building a scalable, easily configurable system, which is fault tolerant as well. Splitting the load reduces hardware related liabilities and at the same time enhances the overall download speed and reliability.

2.1.3 Robots Exclusion Standards

It is a widespread practice to prevent cooperating Web Crawlers from accessing all or part of a website. Some of the policies that a Robot must follow when it is crawling the Web are:

- All the Robots must access the robots.txt file before downloading any files.
- Robots must reveal their identity to the Web servers.
- Robots must minimize the burden of Web servers by using a low retrieval rate and access the Web server only when the server is lightly loaded.

A site owner may wish to instruct the web crawlers by placing a text file by the name of robots.txt in the root of the web site echeleons (e.g. www.martin.edu.my/robots.txt). This text file must contain the instructions in a specific format. A robots.txt file on a website will function as a request that specified robots ignore specified files or directories in their search. Poorly designed robots can lead to serious network and server overload problems. Therefore, all robots must follow the Robot Policies. Also, robots that choose not to follow the Robot Policies might get blacklisted or banned from the website.

2.1.4 Web Crawler Policies

The result of a Web Crawler is normally evaluated depending on how they adhered to the following policies:

- **Selection Policy:** Selection policy refers to the important pages that need to be downloaded first for efficiency purposes considering the huge amount of data on the Web.
- **Re-visit Policy:** Given the dynamic nature of the websites now-a-days, crawling a Web can take a long time, usually in weeks or months. By the time the Web crawler has finished its crawl, many changes could have happened like new contents might have been added or updated or deleted. From the search engine's point of view, there is a cost associated with not detecting an event, and thus keeping an old or outdated copy of a resource. According to J. Cho the most prominently used cost functions are freshness and age.
- **Politeness Policy:** The costs of deploying Web Crawler include network resources, server overload, poorly written robots and also personal robots that disturb the networks and servers. Cho suggests 10 seconds as an interval for access in their studies. Anecdotal evidence from access logs shows that access intervals from known crawlers vary between 20 seconds and 3–4 minutes.
- **Parallelization Policy:** To improve the performance of Web crawler, especially crawling on large data repository like World Wide Web, often technologies like parallel computing are included for the crawlers. A Web crawler can be an agent and by executing multiple agents, along with interaction between the agents, the We can be crawled speedily and more efficiently.[3]

<u>**2.2 Title: Web crawler design Issues: A review**</u> [Deepika, Dr. A. Dixit]

2.2.1 Introduction

This paper thoroughly talks about the functioning of crawlers. Also various types of crawlers are discussed. The paper also talks about several web crawler design issues along with their solutions.

2.2.2 Crawling process

The rudimentary task of a crawler is to fetch pages, parse them to get more URLs, and then fetch pages of these URLs to get even more URLs. It is basically a program that retrieves and stores pages in a repository . These pages are later analysed by a module called indexer. The crawler typically begins with an initial set of URLs called , Seed. Fundamentally, it first places Seed in a queue, where all URLs to be retrieved are stored and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is reiterated until the crawler decides to halt. The working of a typical crawler is shown in fig 2.2.1.

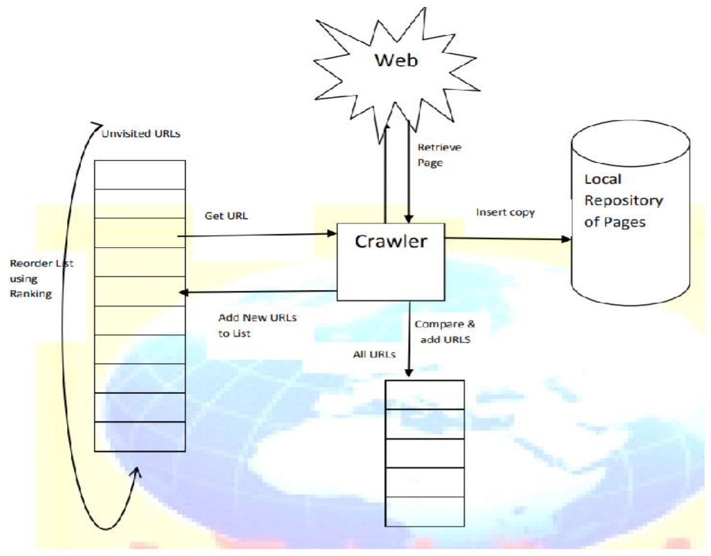


Figure 2.2.1: Working of a Web Crawler

The crawling process is described as follows:

i. The crawler removes the highest-ranked URL from the list of unvisited URLs.
ii. The document is retrieved from the Web.
iii. A copy of the document is placed in the local repository for indexing by the search engine.
iv. The crawler parses the document and extracts the HTML links, with each extracted URL transformed to a standardized format.
v. The extracted URLs are compared to a list of all previously extracted URLs ("All URLs" list in Figure 2.1.1) and any new URL is added to this list.
vi. If the URL is added to the list of all previously extracted URLs, it is also inserted into the list of unvisited URLs for crawling.
vii. The set of URLs is reordered using some scheme such as breadth first ordering or Page Rank, and the process repeats from Step 3 until the set of URLs is empty, the crawl repository is full, or the resources allocated to crawling are exhausted.

2.2.3 Web Crawler Design Issues

- How should the crawler get relevant pages to query?

  The criteria to determine when a page is to be present in a collection are related to the page contents, e.g., words, phrases, etc. [4]However, at times there are other important situations wherein the features of the inner structure of the pages provide a better method to define a collection, than their contents. It is suggested that by knowing the structure of a required page beforehand, desired page(s) can be searched with increased efiiciency. So, rather than fetching all the pages related to search topic, fetch only those pages which will have similar structure as that of sample page in terms of relevancy.

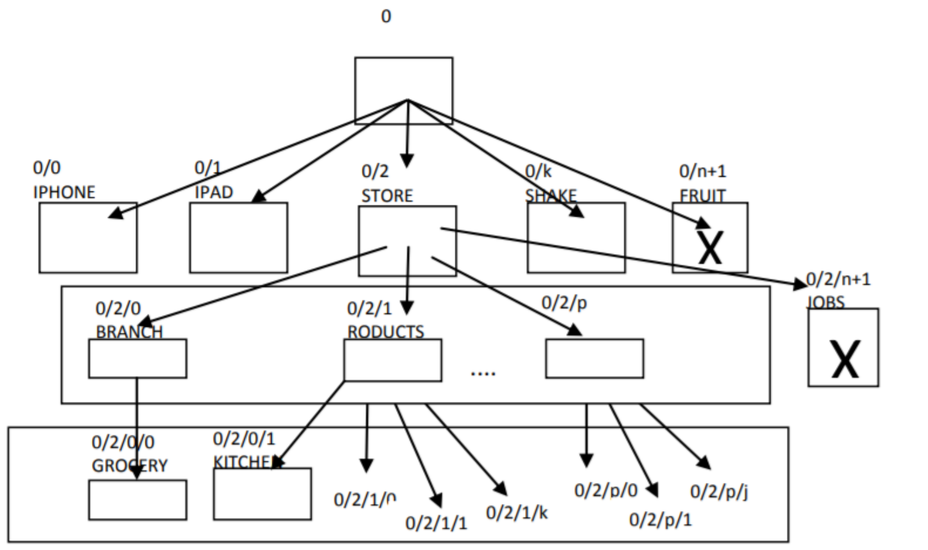


Figure 2.2.2: Sequences of patterns of links

A tool is embellished for fostering structure-driven crawlers which requires minimal effort from users' side as it relies on a sample page of the pages that are to to be fetched. To accomplish this, given a sample page and an entry point to a Web site, the tool greedily traverses the Web site searching for target pages, i.e., pages that are structurally similar to the sample page. Further, it records all paths leading to the target pages and generates a navigation pattern which is composed by sequences of patterns of links a crawler has to follow to reach the target pages as shown in figure 2.2.2. Finally, the tool generates a crawler based on these patterns. From this point on, the crawler can be used to fetch pages that bear structural similarity to the sample page, even if new similar pages are added later.

❖ How should the crawler refresh the page?

Once the crawler has downloaded a significant number of pages, it has to start revisiting those downloaded pages to detect changes and refresh the downloaded collection. Because Web pages are changing at dynamic rates, the crawler needs to carefully decide what page to revisit and what page to skip. There are several approaches to predict when a document is likely to change and therefore require re-crawling such as:

1. Examine how frequently a document has changed in the past. While there are many studies that have examined re-crawl frequency when there is a long period of past change history available, this information is not always available, particularly when a document has only been crawled a few times.
2. Use the Last-Modified and Expires HTTP headers that are returned along with web documents. When they are accurate and properly maintained, these headers can be a very efficient method of improving the freshness of crawled documents. These headers, however, are not always available, and may be deliberately inaccurate to compel web clients to update their locally cached copy of documents.
3. Web pages which are frequently upgraded are detected and accordingly revisit frequency for the pages is dynamically computed.
4. There exists another novel approach for maintaining freshness which uses the anchor text linking documents to determine the likelihood of a document changing, based on statistics gleaned during the current crawl. He shows that this scheme is highly effective when combined with existing stateless crawl ordering schemes.

❖ How should the crawling process be parallelized?

Due to the vastness of the Web, crawlers often run on multiple machines and download pages in parallel. This parallelization is often necessary in order to download a large number of pages in a reasonable amount of time. This generates a continuous stream of new URLs of documents to be downloaded and it is clear that the associated work-load can only be dealed efficiently with proper parallel computing techniques. The incoming new URLs have to be organized by a priority measure in order to download the most relevant documents first.

❖ How should the crawler get time sensitive information?

Usually Search engines crawl the web and take snapshots of site content. Since previous crawls are not archived, hence search results pertain only to a single, recent instant in time.[5] As a result when users request some pages which require past data then search engines are unable to provide because it is not possible to search files that represents snapshot of the web over time.A temporal search engine has been proposed that indexes Internet Archive crawl data to provide search results spanning user specified time ranges. It can generate graphs showing query result hit counts across a given time span and even side-by-side comparisons of different query results. These graphs can be used to, among other things, track a term’s popularity over time for marketing or academic research purposes. A user can input any textual query via a web interface, using either the simple or advanced search functionality as shown in Figure 2.2.3.

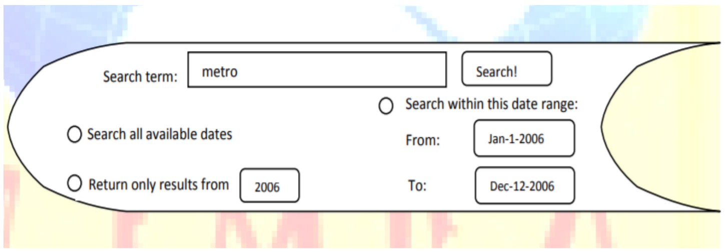

Search term: metro
Search!
Search all available dates
Return only results from 2006
Search within this date range:
From: Jan-1-2006
To: Dec-12-2006

Figure 2.2.3: Chronica’s general search interface

2.3 Title: Choosing Scrapy

2.3.1 Background and Motivation

In this paper, the authors have created an interactive online geographically constrained social services client matching system. In this system, a client would enter basic demographic information and the system would return a set of available social services within a specific geographic area that addressed the clients stated needs. This paper also describes the reason why Scrapy was chosen for this project and explores the viability of scrapy as a powerful and robust tool in computer sciences.

2.3.2 About Scrapy

Scrapy is one of the open source web crawling frameworks that is written in Python. The main purpose of Scrapy is to provide support for web scraping. It was released on June 2, 2008. There are various ongoing Scrapy based projects and many firms or organizations such as CareerBuild, DayWatch, PriceWiki, Tarlabs, etc use Scrapy as a tool to scrape the data from web. The task of capturing and structuring data mined from the web is usually divided into two distinct phases: the crawling and the scraping portions of the task.[6] For this project, Scrapy has been described to be ideal because it is a Python based framework that provides tools for both scraping and crawling. Also, as an open source product, Scrapy has a robust community willing and able to assist new users.

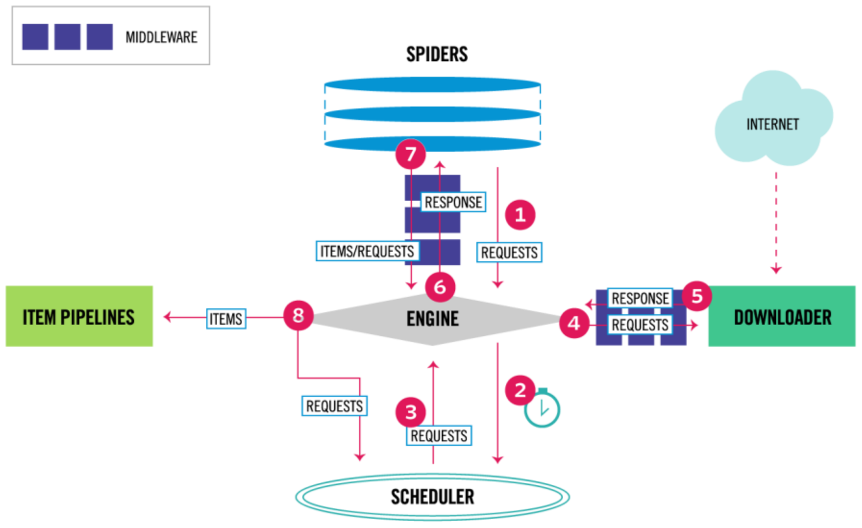


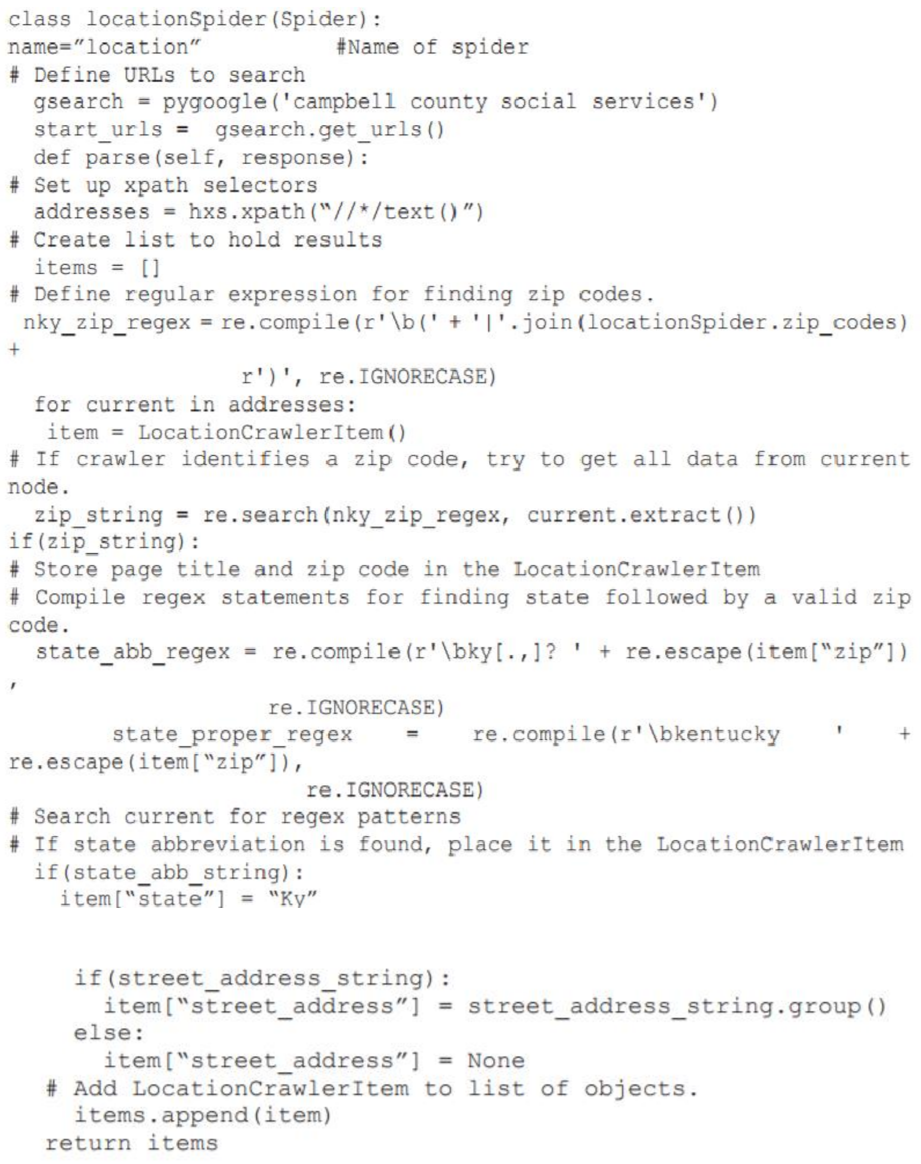
Figure 2.3.1 Architecture of crawler

2.3.3 About the project

The main focus of this project was to write a web crawler that utilizes regular expressions to extract relevant data from geographically targeted websites. The two main tasks for this project were to use Scrapy's spider class to scrape the data and to create a Scrapy item to manage the data.

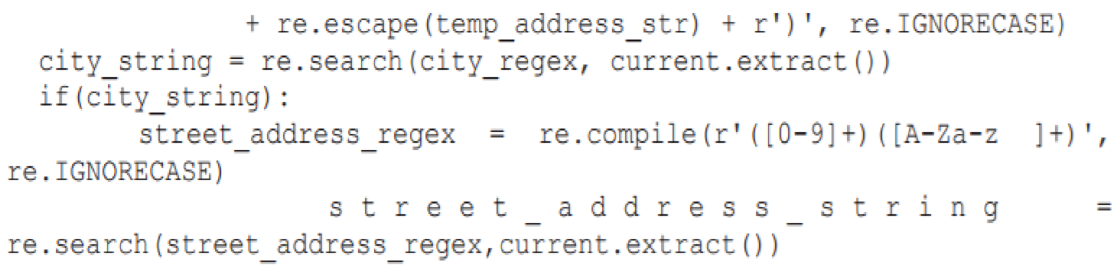
In Scrapy, the spider is the class which defines which web sites will be scrapped, how they will be scrapped, and what data will be collected and placed into the item. Our crawler is designed to extract street addresses from HTML pages.

In the following code, the crawler first looks for a zip code which is in Boone, Kenton or in Campbell countries as defined in an external file. After the zip code is identified, crawler then looks for a string consisting of the state, city and street address.



```
class locationSpider(Spider):
name="location"          #Name of spider
# Define URLs to search
  gsearch = pygoogle('campbell county social services')
  start_urls =  gsearch.get_urls()
  def parse(self, response):
# Set up xpath selectors
  addresses = hxs.xpath("//*/text()")
# Create list to hold results
  items = []
# Define regular expression for finding zip codes.
 nky_zip_regex = re.compile(r'\b(' + '|'.join(locationSpider.zip_codes)
+
                  r')', re.IGNORECASE)
  for current in addresses:
   item = LocationCrawlerItem()
# If crawler identifies a zip code, try to get all data from current
node.
  zip_string = re.search(nky_zip_regex, current.extract())
if(zip_string):
# Store page title and zip code in the LocationCrawlerItem
# Compile regex statements for finding state followed by a valid zip
code.
  state_abb_regex = re.compile(r'\bky[.,]? ' + re.escape(item["zip"])
,
                    re.IGNORECASE)
        state_proper_regex    =    re.compile(r'\bkentucky    '    +
re.escape(item["zip"]),
                       re.IGNORECASE)
# Search current for regex patterns
# If state abbreviation is found, place it in the LocationCrawlerItem
  if(state_abb_string):
    item["state"] = "Ky"


     if(street_address_string):
       item["street_address"] = street_address_string.group()
     else:
       item["street_address"] = None
    # Add LocationCrawlerItem to list of objects.
     items.append(item)
    return items
```



```
               + re.escape(temp_address_str) + r')', re.IGNORECASE)
  city_string = re.search(city_regex, current.extract())
  if(city_string):
        street_address_regex  =  re.compile(r'([0-9]+)([A-Za-z  ]+)',
re.IGNORECASE)
                    street_address_string     =
re.search(street_address_regex,current.extract())
```

Snapshot of the code [7]

After crawling and determining the pages that are to be scraped, the item object was used. In Scrapy, the item defines an object to place the data in. This allows crawler to extract the structured data from unstructured sources.[7] In the example provided, we are seeking to extract all the information necessary to identify a single physical address and place it into the item. The item can later be used to place the data into a database or print the same to a file in JSON, XML or CSV format.

```
class LocationCrawlerItem(Item):
    page_title = Field()
    street_address = Field()
    city = Field()
    state = Field()
    zip = Field()
```

Below is example output after the item has been converted to JSON format and written to a file:

```
[{“city”: “Covington”, “state”: “Kentucky”, “page_title”: [“Welcome To
Kenton County “], “street_address”: “303 Court Street Covington”, “zip”:
“41011"},
{“city”: “Covington”, “state”: “Ky”, “page_title”: [“\r\n\tNorthern
Kentucky Community Action Commission\r\n”], “street_address”: “717
Madison Avenue”, “zip”: “41011"},
{“city”: “Covington”, “state”: “Ky”, “page_title”: [“Kenton County
Attorney's Office :: Departments :: Child Support”], “street_address”:
“41011 “, “zip”: “41011"},
{“city”: “Covington”, “state”: “Ky”, “page_title”: [“\nKenton County
Jail Tracker? - Ask.com\n”], “street_address”: “303 Court Street”,
“zip”: “41011"},
{“city”: “Covington”, “state”: “Ky”, “page_title”: [“\nKenton County
Jail Tracker? - Ask.com\n”], “street_address”: “303 Court Street”,
“zip”: “41011"},
{“city”: “Covington”, “state”: “Ky”, “page_title”: [“Kenton County Homes
\u2013 The Point ARC of N. KY”], “street_address”: “104 West Pike Street
Covington”, “zip”: “41011"},
{“city”: “Newport”, “state”: “Ky”, “page_title”: [“Campbell County
Sheriff”], “street_address”: “1098 Monmouth Street Suite “, “zip”:
“41071"},
{“city”: “Newport”, “state”: “Ky”, “page_title”: [“Campbell County
assistance programs | Newport”], “street_address”: “437 West Ninth
Street”, “zip”: “41071"},
```

Snapshot of the output [7]

2.4 Title: Text Mining Using Python

2.4.1 Introduction

Text mining is the process of analyzing text to cull the information that is useful. As compared to other type of data stored in databases, text is unstructured and very difficult to manage. Text mining is all about finding unknown information that tries to extract pattern from large databases.[8] Approximately 90% of the world's data, generated daily is in unstructured formats. Text is just raw data. Text mining generally refers to analyzing large natural language texts and detect usage patterns to extract useful information.

2.4.2 Basics of Text Mining

Information retrieval is the first step in text mining. It is the process of searching and recovering information from huge amount of stored data. Specific rules are there to recoup information from text and those rules also defines what degree of retrieval should be considered. This is process of identifying and returning relevant information. Text categorization is text mining operation. It is the process of assigning the document to predefined categories. Collected documents should be classified as per the requirement. Information extraction is also part of text mining process. Information extraction includes identification and extraction of certain entities in text and representation of it in required format. Important task of text mining is to search for structured data inside the document. Text mining also includes identification of trends in data.

2.4.3 Text Mining vs Data Mining

Data mining is one step in the whole process of knowledge discovery from data. Knowledge from data process involves collection of important, rational, new, and useful knowledge from data. In text mining there is linguistic analysis and the main focus of search engines is on text search that notably focus on text based web content. The text data is free-form, unstructured and semi-structured data. Text may include reports, articles, emails, letters, etc. In data mining, data can be in the form of images, audio, videos, etc. Data mining involves case-based reasoning, data visualization. Also the main use of data mining includes cross-selling, segmentation and profiling, response modeling.

2.4.4 Python functionalities for Text Mining

For extracting and parsing contents, python provides regular expression handling in the re module and Beautiful Soup. Also, there are specialized module for JSON feeds and XML. Scrapy is a huge framework for crawling and extraction. The NLTK package contains copious natural language processing methods say sentence and word tokenization, part of speech tagging, chunking and classification. Natural Language Toolkit is a retinue of libraries and programs that helps in analytical NLP in python language.[9] It provides easy to use interface to text processing libraries for classification, tokenization and parsing. NLTK is available for windows, MAC OS and Linux.

2.4.5 Applications of Text Mining using Python

Text mining is used in many sectors for reviewing large data sets. In media and publishing sectors, text mining is mainly used for production and accrual for information retrieval that is done by extracting, loading, and transporting the information. In the Banks and Financial market, CRM tool is used for to improve the customer communication management by resending the message by search engine asking question in natural language. Hospital and Pharmaceutical firms use classification and extraction of information from articles, scientific abstracts and patents.[10] Various type of applications are used is in the telecommunications industries: the most important aim of these industries are that all applications find an answer, from human resources management to market analysis, from customer opinion survey to spelling correction.

2.5 <u>Title: Big Data – A Brief Study</u> [Ravi Narasimhan, Bhuvaneshwari T]

In this research article, the author has briefed the current industry lingo called Big Data and has covered the components of big data from the perspective of Hadoop. The study will help to have a comprehensive understanding of big data and its various components in hadoop framework.

2.5.1. Introduction

As an inclusive definition, Big Data as we see, is something so huge and mosaic that it is impossible for traditional systems and data-warehousing tools to process and work on them. Big data can neither be worked in consequent to the traditional SQL like queries nor can the RDBMS be used for storage purpose. Therefore, a wide variety of scalable database tools and techniques have evolved. Hadoop, an open source distributed data processing system (which includes HIVE data warehouse system and HBase database) is one of the prominent and well known solutions. NoSQL has ameliorated protrusion as a non-relational database with the inclusion of MongoDB, DynamoDB from Amazon and Cassandra from Apache.

Day after day, trillions and zillions of data is generated all over the universe which includes machine generated, human generated data and other that are generated by nature. Machine generated data can come from Airplanes, Cars, CERN systems (particle physics Lab-Hadron collider) and likewise. Human generated data comprises of data retrieved from social media like facebook, twitter, etc. Nature generated data comes from whether, genome sequencing, deep ocean data, the universe data, etc.

2.5.2. Big Data Characteristics – The 5 V's

Big Data is peculiarized by the 5 V's – Volume, Velocity, Variety, Veracity and Value. In addition we can also include Variability.

A) Volume

The size of data has increased from Megabytes and Gigabytes to Terabytes, Petabytes and Exabytes. Over the past few years, because of the super low-price in hardware and storage memory costs, it has become cost-effective to the store data. This has resulted in storage of huge amount of Structured as well as Unstructured data from various sources, like data from social media, sensors, machine-to-machine data, from airplanes, from digitalization of books, etc. To store this huge amount of data, we have various databases including Green-plum and Hadoop. The storage of data in Hadoop is in the form Distributed File System called HDFS which makes Data available to multiple computing nodes.[11] Usage pattern consist of the following three steps:

1. Loading data into HDFS
2. MapReduce Operations
3. Retrieval of results from HDFS

It is more of a batch processing and hence suited for analytical and statistical purposes.

B) Velocity

Data is being generated at an increasing rate. Users want the full data to be given to them almost instantly. Now-a-days data-flow is immense and continuous. Data is coming to us at an unprecedented speed. Streaming data has made it possible to have a real time catering of data.

With so much of data being generated at a very high speed, the terminology called streaming data came into headland. It’s not only about incoming data, but also about insights and the decisions that are needed to be taken at an equally high pace.

C) Variety

Data to become Big Data comes in from various sources. Data is mainly divided into three different types –

- **Structured Data** is what the traditional Database management systems work with. It consist of Rows and Columns and re-sides in fixed fields in a file.
- **Semi-Structured Data** does not confirm to a specific arrangement but consists of Ttage to separate the data elements.
- **Un-Structured Data** does not have and does not adhere to a pre-defined data like representation. E-mails, video and audio are some of the unstructured data files.

D) Veracity

As Big Data becomes bigger and the multiple sources of big data are ever increasing, there is lot of chances that there is huge inconsistency and abnormality in the Data.[12] The reliability or the trustworthiness of the data becomes questionable. Data inconsistency and abnormality is one major challenging factor that is a part and parcel of Big Data.

E) Value

Cost is one of the major factors that all organizations need to look into when it comes to implementing big data, or for the same any software package or framework implementation. The initiative of entering into Big Data is very cynical and the value needs to be premeditatedly cogitated in value to price perspective.

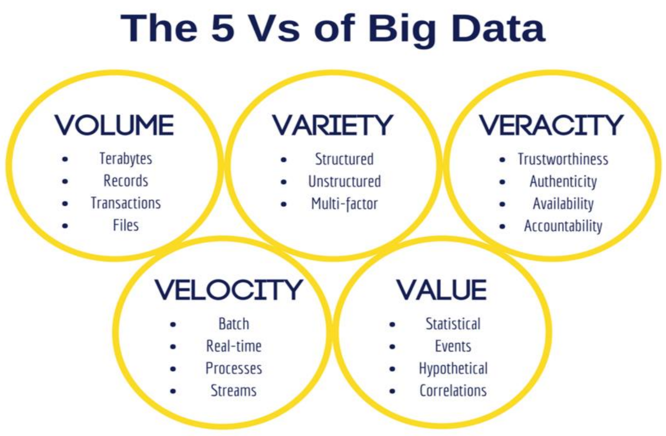


Figure 2.5.1 The 5 Vs of Big data

2.5.3. Big Data Components on Hadoop Framework

Apache Hadoop is a software library, an open-source framework for reliable, scalable and distributed computing. It is designed to cope with distributed computing of complex and large datasets dealing with multiple clusters of computers using simple programming models. Created by Doug Cutting and Mike Caferella in the year 2005, it is name after a toy elephant. Scaling up from Single servers to thousands of machines with Local storage and computation are the advantage that Hadoop offers.[13] This is one of the major advantages that Hadoop offers as we can use in-expensive hardware.

Hadoop Modules:

- **Hadoop Distributed File System (HDFS)**
  A distributed file system which helps to store huge and complex amount of data in a reliable format providing fault tolerant file system. HDFS follows the structure of Master and Slave, in which we have one or more devices called as slave devices that are controlled by one device that is called as master device. It is a java based file system.

- **Hadoop YARN / MapReduce**

  MapReduce/YARN is a cluster recource management and has been built as a programming model in the Hadoop framework to process large amounts of data in a distributed & parallel environment on a cluster. Yarn is the heart of Hadoop and ResourceManager and NodeManager being help to manage the applications in a distributed manner.

- **HBase**

  HBASE is a Hadoop database which is a non-relational (NoSQL) database that runs on the top of HDFS. HBASE allows random, real time read and write access for the big data, is columnar, provides fault tolerant storage and fast access. It also provides transactional capabilities by allowing updates, insertions, deletions etc. Tables in HBase can be used to serve as inputs or outputs for jobs running in MapReduce.

  Some more components are listed below-

- Pig
- Hive
- Sqoop
- ZooKeeper
- Avro
- Cassandra
- Mahout
- Spark
- Flume
- R

2.6 A review of programming languages for web scraping

The foremost job of any researcher for the task of data extraction is the choice of too and language. This paper talks about the advantages and disadvantages of four programming languages: C, Java, PHP and Python vis-à-vis their libraries and several programming features offered by these languages in the arena of scraping websites and extraction of data. Also, the related libraries along with the methods of their utilization have also been discussed in brief.

2.6.1. Significance of research

As the research furthered, it became clear to the author that each tool carries some positive as well as negative aspect. The selection of the language or tool for scraping a website depends on more than one factor. The most essential ones being the purpose of extracting data from a particular website, the deciding parameters that are to be used for scraping and the programming language with which the individual is familiar and is comfortable working in.[14] The author has expressed that at one stage of work, one language may be suitable and at some other stage, another language may seem more feasible. This research will aid an individual to understand which language among C/C++, JAVA, PHP and Python is more suitable to carry out web scraping and extraction. After getting to know the pros and cons of employing a particular language, the user will also be able to choose a different language for needs at various stages of the research.

2.6.2. Discussions

Four options have been discussed in this paper. The first is native code – C/C++, then a very powerful object oriented programming language of today's times, namely Java, the third is a scripting language PHP and lastly a contemporary and dynamically typed language Python.

2.6.2.1 Using the native language C/C++

The author has identified three steps for extracting data using the native language:

1. Using the library "libcurl", download the html page. The libcurl library supports HTTP and HTTPS besides other protocols such as SFTP, Telnet, FTP, SMTP and some more. At present, the library runs in the same manner on many platforms like Windows, Solaris, Linux, Symbian, Android, macOS, Blackberry, Darwin etc. and hence it would be safe to say that this library is absolutely portable. This library also supports Kerberos. It is IPv6 compatible and freely available. In Linux, the command employed is :

```
$ curl <URL>
```

This command downloads the page and by default the terminal is the output but a file can be specified to store the data. From Linux, the following command can be utilized:

```
$ curl<URL>>><filename>
```

2. Next step is converting"the downloaded HTML page to a decently formatted and more readable XML format using the "libtidy" library. This library is a static and dynamic C library which can be integrated with many languages. The code using libtidy goes through the following steps:

- Initializing document using tidyCreate method
- Converting it into xml by using tidyOptSetBool method
- Capturing diagnostics using tidySetErrorBuffer
- Parsing the input using tidyParseString
- Tidying up the things using tidyCleanAndRepair function
- Printing if required

4. The last step is parsing the xml document using "libxml" library. This library is capable of working on Linux, Windows as well as macOS but it works best on Linux. It requires only ANSI C API dependency. It uses libz and iconv libraries during is configuration. Using these libraries, one can write code to parse an xml page and select the needed

content. Coming to the pros of employing C/C++, it provides more flexibility and control to the user. The low level work ensures speed and reliability. But there are some disadvantages too. It involves tedious task of writing huge sections of code and the learning process is tougher. The above mentioned three steps have to be taken care of separately.[15] Every time a program is modified, it has to be compiled. Memory management and string handling are two serious issues.

2.6.2.2 Using object oriented language JAVA

There are many ways available to scrape data in Java. The author has highlighted two main libraries in this respect. One is HtmlUnit and other is Jsoup. Jsoup is used to parse HTML pages and uses CSS, JQuery and DOM for this purpose. The general course of steps involves scraping and parsing HTML document from a file or a URL. Next is extracting the required data by employing CSS or other Selectors. There are many inbuilt methods like getElementByClass(String className), getElementById(String id), getElementByTag(String Tag), getElementByAttribute(String attribute). Several other methods that may be used to parse and do manipulations on data are: tagname that finds elements using a tag – #id, .class, [attribute] etc. HtmlUnit is fundamentally a browser without a Graphical User Interface. It provides APIs allowing someone to click links, fill forms and other facilities. Some get methods can be used to search for a particular element. Or one can also use Xpath to query elements. One can easily submit a form by following the steps: go to the required page, get the form and find the submit button and the field that is to be changed or submitted, modify the field and submit the form by clicking the submit button. One can also specify a proxy server if one needs to connect through one. Another tool is WebDriver like Selenium. Although it is used typically in automated web application testing, it's APIs can be used to extract desired information. The merits of using Java are it is a widely used language and many powerful ibraries are freely available. Eclipse and Netbeans are powerful IDEs that help analyze and organize code.[16] A specific web browser can be simulated and the real success of using Java is realized in big projects. Pages having diversed content can be easily scraped. Coming to demerits, it is difficult to code in Java and the outputs are unusually detailed. After every modification, it is mandatory to compile. The powerful IDEs consume lots of machine resources. One has to wait for the complete webpage to be loaded as only then the elements can be accessed. As a result of simulation of web browsers, many unnecessary files get loaded creating heavy traffic.

2.6.2.3 Using the scripting language PHP

To parse and scrape data in PHP, one needs the PHP Simple HTML DOM Parser which is written in PHP itself. One can select tags from an HTML page just like jquery. Contents can be extracted from the HTML page in a single line. The steps to obtian HTML elements are:

Creating COM from a file or URL using the file_get_html method which takes URL or the file name from which data is to be extracted as an argument.

Other functions such as find('img') can be utilized for extracting images and find('a') for extracting links. Many elements can be altered by creating a DOM first and then finding a specific 'div' and then putting the required text in the proper place. The function file_get_html() will dump all the contents of the web page as a plain text. Data can be scraped easily by running a loop and selecting each 'div' element and converting its contents into plaintext, storing them as a string and showing them when required. Numerous other things can be performed using a simple php library "simple_html_dom.php". Advantages of using PHP include it's free availability, it's scripting nature, easy coding, acceptance by almost all enterprises and the requirement of just one single file as the necessary library. Coming to the negative side, PHP has no support for parallel works. Also, there is no way for managing memory.[17] There is no compatibility for Unicode. It is not easy to debug. To implement security checks in PHP, the functions that would be used have very long names. For every web request, whole of the script is re-initialised and run from the starting.

2.6.2.4 Using the dynamic language Python

Python offers many powerful libraries for scraping websites like Beautiful Soup, Requests, lxml, Selenium and Scrapy. Requests can be used to not only extract raw HTML but also to

post to forms and access APIs. Beautiful Soup is an amazing package if one wants to parse and select the desired data. Next is lxml which does scraping speedily. It works on Xpath and CSS selectors. It is preferred over Beautiful Soup in scenarios where the documents are untidy. Next is Selenium which is extremely useful where Javascript is involved. Selenium actually simulates Web Browser and can enable one click on links and enter data into forms. However, if one requires to do all of the above and build a complete spider that can crawl through websites in a very systematic way then the best thing to use may be Scrapy. Scrapy is written in Python and it depends on some Python packages: lxml, which is an excellent HTML and XML parser, parsel, which is written on top of lxml and facilitates data extraction. W3lib assists in handling webpage encoding and URL. A library called twisted which is an asynchronous networking framework. To deal with network security, there are libraries called cryptography and pyOpenSSL.[18] One can write spiders using Scrapy. Spiders are Python classes that set the definitions of the web scraping techniques. In these classes the ways of parsing and crawling web pages are defined. The work of the spiders generally go through a four step process: first, initial requests are generated to crawl URLs and a callback function is specified to handle the response from these requests. Next the downloaded response is parsed and data extracted. Thirdly page contents are parsed using selectors and items are generated using the parsed data. Finally items that the spider class return shall be stored in a proper database or a data warehouse as per the requirement.

There are many advantages of using Python like object oriented facilities are available but are not mandatory to use. Similar is the case with exceptions. Python is easy to understand and intuitive like English. It enforces indentation which is a good programming practice. It offers an outstanding set of powerful libraries. Also, a few lines of code can tackle a huge problem.

More time can be spent on problem solving and designing algorithms rather than getting delayed with programming language features. There are many built-in datatypes that are extremely useful like lists, sequences, functions and dictionaries. Development time is very fast. It is possible to compile and run the program till an error pops up. Native languages like C can be integrated into Python. There is huge community support. One disadvantage is

that being dynamically types, it may become difficult to keep track of variables and other things.

2.6.3 Conclusion

The author feels that the most appropriate and efficient language that can be used for extracting data from software repositories and analyze the trends in software projects is Python . In all probability, it is most apt to use Scrapy framework as it offers excellent functionalities for data mining and also its proper storage and use. If simulation is requied, then Selenium would be best suited as it can enable logging into websites if required.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 System Design

3.1.1 Hardware Requirements

The hardware requirements for this project to be implemented are-

- Processor: Intel(R) Core(TM) i5-5200U CPU @2.20GHz 2.20GHz
- RAM: 8GB
- 64-bit Operating System

3.1.2 Software Requirements-

In this project, the softwares that required for implementation includes

- **Anaconda Prompt-** Conda Prompt includes inbuilt Python packages.
  It is used to run Python programmes.
- **Spyder-** Spyder is an open-source cross-platform integrated development environment for scientific programming in Python.

3.1.3 Algorithms used

- Scraping Website
- Data Cleaning and processing
- Ranking Algorithm

3.1.3.1 Scraping Website

With the incessant expansion of internet, it has become important to retrieve data speedily and accurately. This task would be impossible to carry out if we lack effective “search engines”. To achieve this goal, a software called ‘Web Crawler’ is exploited which can be run using various algorithms. Spider is a class that defines the way how a particular site will be scraped, how the links would be followed i.e. how the crawl would be performed and how the data would be extracted. CrawlSpider is a Spider that defines a set of rules for the crawler to follow the links and scrap multiple pages. CrawlSpider class typically uses Link Extractors but it can be used in any spider, even by foregoing to subclass from CrawlSpider. Link Extractors are objects that can be used as default or can be customized to extract links from web pages. In the scenario of our problem statement, we have written a fundamental crawler program which takes a seed URL to begin the search with and returns a list of URLs related to the three ranking parameters.

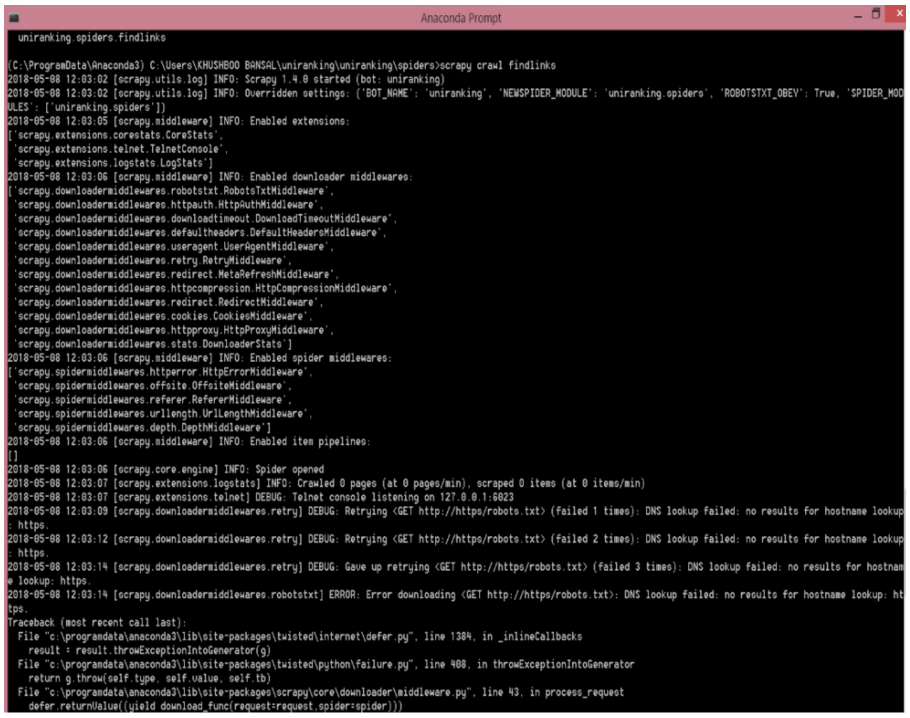
Following steps of the scraping cycle are common to (more or less) any kind of spider:

1. The process starts by generating initial requests, which are obtained by calling the start_requests() function, to crawl the URLs specified in start_urls and then specifying a callback method i.e parse method which is to be called with the response downloaded from those requests.
2. The response i.e. the web page is analyzed in the callback function which returns either dictionaries of python in extracted data, Request objects, Item objects, or an iterable of such objects.
3. Next, the contents of the web pages are parsed by the callback method by generally using Selectors and items are generated with the parsed data.
4. At the end, the items that are returned by the spider are stored in a database or written to a file.

Following is a pseudo code for scraping data from the site 'data.gov.in':

```
1. import scrapy
2. from scrapy.spiders, import CrawlSpider, Rule
3. import LinkExtractor from scrapy.linkextractors
4. defne a class by the name UniRanking and make it inherit CrawlSpider class
5. specify the allowed_domains as www.data.gov.in
```

6. specify start_urls as https://data.gov.in/resources/
7. set scholarship, student enrollment, teachers/faculty as allow attribute, parse_item as callback in the Rules class
8. define the parse_item method extract the link in an item object



```
 uniranking.spiders.findlinks

(C:\ProgramData\Anaconda3) C:\Users\KHUSHBOO BANSAL\uniranking\uniranking\spiders>scrapy crawl findlinks
2018-05-08 12:03:02 [scrapy.utils.log] INFO: Scrapy 1.4.0 started (bot: uniranking)
2018-05-08 12:03:02 [scrapy.utils.log] INFO: Overridden settings: {'BOT_NAME': 'uniranking', 'NEWSPIDER_MODULE': 'uniranking.spiders', 'ROBOTSTXT_OBEY': True, 'SPIDER_MOD
ULES': ['uniranking.spiders']}
2018-05-08 12:03:05 [scrapy.middleware] INFO: Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.logstats.LogStats']
2018-05-08 12:03:06 [scrapy.middleware] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
 'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware',
 'scrapy.downloadermiddlewares.retry.RetryMiddleware',
 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware',
 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware',
 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware',
 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware',
 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware',
 'scrapy.downloadermiddlewares.stats.DownloaderStats']
2018-05-08 12:03:06 [scrapy.middleware] INFO: Enabled spider middlewares:
['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware',
 'scrapy.spidermiddlewares.offsite.OffsiteMiddleware',
 'scrapy.spidermiddlewares.referer.RefererMiddleware',
 'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware',
 'scrapy.spidermiddlewares.depth.DepthMiddleware']
2018-05-08 12:03:06 [scrapy.middleware] INFO: Enabled item pipelines:
[]
2018-05-08 12:03:06 [scrapy.core.engine] INFO: Spider opened
2018-05-08 12:03:07 [scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2018-05-08 12:03:07 [scrapy.extensions.telnet] DEBUG: Telnet console listening on 127.0.0.1:6023
2018-05-08 12:03:09 [scrapy.downloadermiddlewares.retry] DEBUG: Retrying <GET http://https/robots.txt> (failed 1 times): DNS lookup failed: no results for hostname lookup
: https.
2018-05-08 12:03:12 [scrapy.downloadermiddlewares.retry] DEBUG: Retrying <GET http://https/robots.txt> (failed 2 times): DNS lookup failed: no results for hostname lookup
: https.
2018-05-08 12:03:14 [scrapy.downloadermiddlewares.retry] DEBUG: Gave up retrying <GET http://https/robots.txt> (failed 3 times): DNS lookup failed: no results for hostnam
e lookup: https.
2018-05-08 12:03:14 [scrapy.downloadermiddlewares.robotstxt] ERROR: Error downloading <GET http://https/robots.txt>: DNS lookup failed: no results for hostname lookup: ht
tps.
Traceback (most recent call last):
  File "c:\programdata\anaconda3\lib\site-packages\twisted\internet\defer.py", line 1384, in _inlineCallbacks
    result = result.throwExceptionIntoGenerator(g)
  File "c:\programdata\anaconda3\lib\site-packages\twisted\python\failure.py", line 408, in throwExceptionIntoGenerator
    return g.throw(self.type, self.value, self.tb)
  File "c:\programdata\anaconda3\lib\site-packages\scrapy\core\downloader\middleware.py", line 43, in process_request
    defer.returnValue((yield download_func(request=request,spider=spider)))
```

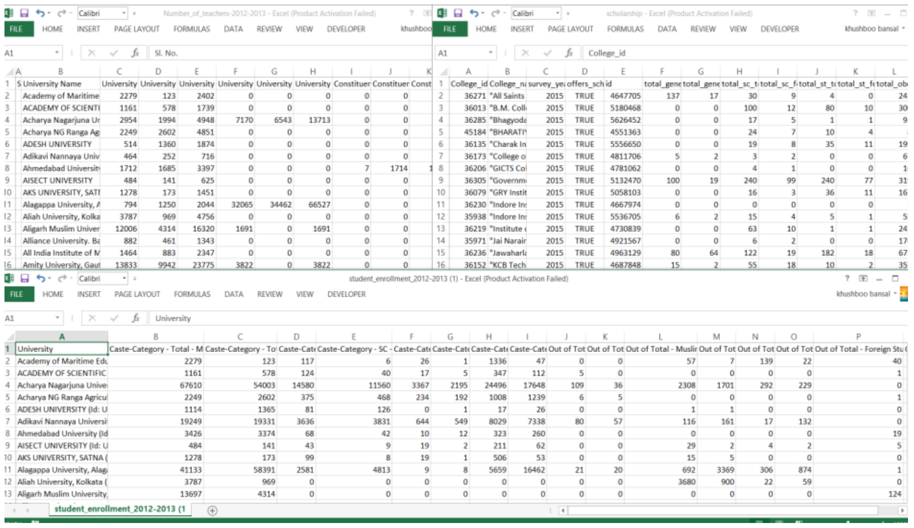
Fig 3.1.3.1 Running Spider on Anaconda Prompt

2. Data cleaning and processing

This particular step is very crucial for the success of our final ranking as it eliminates the undesirable data and produces a single file for us to run our sorting algorithm on. Coming to cleaning the datasets obtained after crawling the government website. Here, the first step would be to remove the unnecessary punctuation marks like dot, inverted commas, commas, colons and white spaces. Then to bring about uniformity, the datasets are converted into lower case and sorted alphabetically in ascending order. The extraneous columns are deleted the relevant ones are added for each row. Next, the data in the three datasets are compared as per the names of the colleges and only the required columns are included in the new dataframe which is written in a fresh excel file.

Steps:

1. Read the excel files into separate data frames using pandas.DataFrame and pandas.read_csv function.
2. Apply str.replace() function to get rid of white spaces and punctuation marks in the dataframes.
3. Apply str.lower() function to the columns containing college names.
4. Sort the datasets as per college/university names using DataFrame.sort_values function.
5. Sum the informative columns and delete the unwanted ones using list.remove() and Series.sum() functions.
6. Compare the student enrollment and scholarship datasets by iterating through the rows using pandas.DataFrame.iterrows() function and store the values in columns having common college names in a separate dataframe. The college name in each row is split into list of its comprising words using str.split() and the first two words are matched in the respective columns in both the datasets.
7. Compare this new dataframe with faculty dataframe following the same procedure as in the above step.
8. Merge the dataframes obtained in steps 6 and 7 and finally apply the ranking algorithm to it.



Number_of_teachers-2012-2013 - Excel (Product Activation Failed)

	A	B	C	D	E	F	G	H	I	J	K
1	S	University Name	University	University	University	University	University	University	Constituer	Constituer	Const
2		Academy of Maritime	2279	123	2402	0	0	0	0	0	
3		ACADEMY OF SCIENTI	1161	578	1739	0	0	0	0	0	
4		Acharya Nagarjuna Ur	2954	1994	4948	7170	6543	13713	0	0	
5		Acharya NG Ranga Ag	2249	2602	4851	0	0	0	0	0	
6		ADESH UNIVERSITY	514	1360	1874	0	0	0	0	0	
7		Adikavi Nannaya Univ	464	252	716	0	0	0	0	0	
8		Ahmedabad University	1712	1685	3397	0	0	0	7	1714	1
9		AISECT UNIVERSITY	484	141	625	0	0	0	0	0	
10		AKS UNIVERSITY, SATI	1278	173	1451	0	0	0	0	0	
11		Alagappa University, A	794	1250	2044	32065	34462	66527	0	0	
12		Aliah University, Kolka	3787	969	4756	0	0	0	0	0	
13		Aligarh Muslim Univer	12006	4314	16320	1691	0	1691	0	0	
14		Alliance University. Ba	882	461	1343	0	0	0	0	0	
15		All India Institute of M	1464	883	2347	0	0	0	0	0	
16		Amity University, Gaut	13833	9942	23775	3822	0	3822	0	0	

scholarship - Excel (Product Activation Failed)

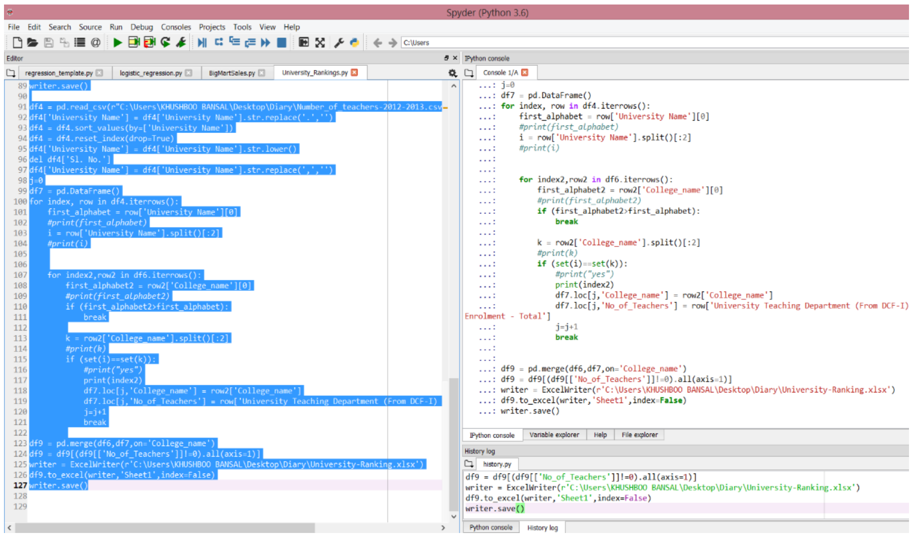
	A	B	C	D	E	F	G	H	I	J	K	L
1	College_id	College_na	survey_yea	offers_sch	id	total_gene	total_gene	total_sc_t	total_sc_f	total_st_t	total_st_f	total_ob
2	36271	"All Saints	2015	TRUE	4647705	137	17	30	9	4	0	24
3	36013	"B.M. Coll	2015	TRUE	5180468	0	0	100	12	80	10	30
4	36285	"Bhagyoda	2015	TRUE	5626452	0	0	17	5	1	1	9
5	45184	"BHARATI	2015	TRUE	4551363	0	0	24	7	10	4	
6	36135	"Charak In	2015	TRUE	5556650	0	0	19	8	35	11	19
7	36173	"College o	2015	TRUE	4811706	5	2	3	2	0	0	6
8	36206	"GICTS Col	2015	TRUE	4781062	0	0	4	1	0	0	1
9	36305	"Governm	2015	TRUE	5132470	100	19	240	99	240	77	31
10	36079	"GRY Instit	2015	TRUE	5058103	0	0	16	3	36	11	16
11	36230	"Indore Ins	2015	TRUE	4667974	0	0	0	0	0	0	
12	35938	"Indore Ins	2015	TRUE	5536705	6	2	15	4	5	1	5
13	36219	"Institute c	2015	TRUE	4730839	0	0	63	10	1	1	24
14	35971	"Jai Narair	2015	TRUE	4921567	0	0	6	2	0	0	17
15	36236	"Jawaharla	2015	TRUE	4963129	80	64	122	19	182	18	67
16	36152	"KCB Tech	2015	TRUE	4687848	15	2	55	18	10	2	35

student_enrollment_2012-2013 (1) - Excel (Product Activation Failed)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	University	Caste-Category - Total - M	Caste-Category - To	Caste-Cate	Caste-Category - SC -	Caste-Cate	Caste-Cate	Caste-Cate	Caste-Cate	Out of Tot	Out of Tot	Out of Total - Muslir	Out of Tot	Out of Tot	Out of Tot	Out of Total - Foreign Stu
2	Academy of Maritime Edu	2279	123	117	6	26	1	1336	47	0	0	57	7	139	22	40
3	ACADEMY OF SCIENTIFIC	1161	578	124	40	17	5	347	112	5	0	0	0	0	0	1
4	Acharya Nagarjuna Unive	67610	54003	14580	11560	3367	2195	24496	17648	109	36	2308	1701	292	229	0
5	Acharya NG Ranga Agricul	2249	2602	375	468	234	192	1008	1239	6	5	0	0	0	0	1
6	ADESH UNIVERSITY (Id: U	1114	1365	81	126	0	1	17	26	0	0	1	1	0	0	0
7	Adikavi Nannaya Universi	19249	19331	3636	3831	644	549	8029	7338	80	57	116	161	17	132	0
8	Ahmedabad University (Id	3426	3374	68	42	10	12	323	260	0	0	0	0	0	0	19
9	AISECT UNIVERSITY (Id: U	484	141	43	9	19	2	211	62	0	0	29	2	4	2	5
10	AKS UNIVERSITY, SATNA (	1278	173	99	8	19	1	506	53	0	0	15	5	0	0	0
11	Alagappa University, Alag	41133	58391	2581	4813	9	8	5659	16462	21	20	692	3369	306	874	1
12	Aliah University, Kolkata (	3787	969	0	0	0	0	0	0	0	0	3680	900	22	59	0
13	Aligarh Muslim University,	13697	4314	0	0	0	0	0	0	0	0	0	0	0	0	124

student_enrollment_2012-2013 (1

Fig 3.1.3.2 The datasets obtained after crawling



Spyder (Python 3.6)

regression_template.py | logistic_regression.py | BigMartSales.py | University_Rankings.py

```
writer.save()

df4 = pd.read_csv(r"C:\Users\KHUSHBOO BANSAL\Desktop\Diary\Number_of_teachers-2012-2013.csv
df4['University Name'] = df4['University Name'].str.replace('.','')
df4 = df4.sort_values(by=['University Name'])
df4 = df4.reset_index(drop=True)
df4['University Name'] = df4['University Name'].str.lower()
del df4['Sl. No.']
df4['University Name'] = df4['University Name'].str.replace(',','')
j=0
df7 = pd.DataFrame()
for index, row in df4.iterrows():
    first_alphabet = row['University Name'][0]
    #print(first_alphabet)
    i = row['University Name'].split()[:2]
    #print(i)


    for index2,row2 in df6.iterrows():
        first_alphabet2 = row2['College_name'][0]
        #print(first_alphabet2)
        if (first_alphabet2>first_alphabet):
            break

        k = row2['College_name'].split()[:2]
        #print(k)
        if (set(i)==set(k)):
            #print("yes")
            print(index2)
            df7.loc[j,'College_name'] = row2['College_name']
            df7.loc[j,'No_of_Teachers'] = row['University Teaching Department (From DCF-I)
            j=j+1
            break

df9 = pd.merge(df6,df7,on='College_name')
df9 = df9[(df9[['No_of_Teachers']]!=0).all(axis=1)]
writer = ExcelWriter(r'C:\Users\KHUSHBOO BANSAL\Desktop\Diary\University-Ranking.xlsx')
df9.to_excel(writer,'Sheet1',index=False)
writer.save()
```

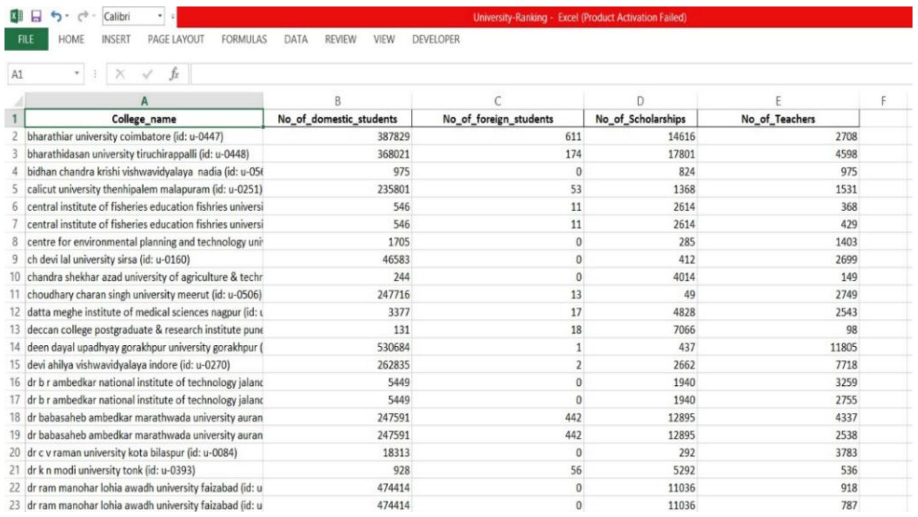
IPython console — Console 1/A

```
   ...: j=0
   ...: df7 = pd.DataFrame()
   ...: for index, row in df4.iterrows():
   ...:     first_alphabet = row['University Name'][0]
   ...:     #print(first_alphabet)
   ...:     i = row['University Name'].split()[:2]
   ...:     #print(i)
   ...:
   ...:
   ...:     for index2,row2 in df6.iterrows():
   ...:         first_alphabet2 = row2['College_name'][0]
   ...:         #print(first_alphabet2)
   ...:         if (first_alphabet2>first_alphabet):
   ...:             break
   ...:
   ...:         k = row2['College_name'].split()[:2]
   ...:         #print(k)
   ...:         if (set(i)==set(k)):
   ...:             #print("yes")
   ...:             print(index2)
   ...:             df7.loc[j,'College_name'] = row2['College_name']
   ...:             df7.loc[j,'No_of_Teachers'] = row['University Teaching Department (From DCF-I)
Enrolment - Total']
   ...:             j=j+1
   ...:             break
   ...:
   ...:
   ...: df9 = pd.merge(df6,df7,on='College_name')
   ...: df9 = df9[(df9[['No_of_Teachers']]!=0).all(axis=1)]
   ...: writer = ExcelWriter(r'C:\Users\KHUSHBOO BANSAL\Desktop\Diary\University-Ranking.xlsx')
   ...: df9.to_excel(writer,'Sheet1',index=False)
   ...: writer.save()
```

History log — history.py

```
df9 = df9[(df9[['No_of_Teachers']]!=0).all(axis=1)]
writer = ExcelWriter(r'C:\Users\KHUSHBOO BANSAL\Desktop\Diary\University-Ranking.xlsx')
df9.to_excel(writer,'Sheet1',index=False)
writer.save()
```

Fig 3.1.3.3 Python console in Spyder framework performing cleaning on the datasets



	A	B	C	D	E
1	College_name	No_of_domestic_students	No_of_foreign_students	No_of_Scholarships	No_of_Teachers
2	bharathiar university coimbatore (id: u-0447)	387829	611	14616	2708
3	bharathidasan university tiruchirappalli (id: u-0448)	368021	174	17801	4598
4	bidhan chandra krishi vishwavidyalaya nadia (id: u-05	975	0	824	975
5	calicut university thenhipalem malapuram (id: u-0251)	235801	53	1368	1531
6	central institute of fisheries education fishries universi	546	11	2614	368
7	central institute of fisheries education fishries universi	546	11	2614	429
8	centre for environmental planning and technology uni	1705	0	285	1403
9	ch devi lal university sirsa (id: u-0160)	46583	0	412	2699
10	chandra shekhar azad university of agriculture & techr	244	0	4014	149
11	choudhary charan singh university meerut (id: u-0506)	247716	13	49	2749
12	datta meghe institute of medical sciences nagpur (id: u	3377	17	4828	2543
13	deccan college postgraduate & research institute pune	131	18	7066	98
14	deen dayal upadhyay gorakhpur university gorakhpur (	530684	1	437	11805
15	devi ahilya vishwavidyalaya indore (id: u-0270)	262835	2	2662	7718
16	dr b r ambedkar national institute of technology jalanc	5449	0	1940	3259
17	dr b r ambedkar national institute of technology jalanc	5449	0	1940	2755
18	dr babasaheb ambedkar marathwada university auran	247591	442	12895	4337
19	dr babasaheb ambedkar marathwada university auran	247591	442	12895	2538
20	dr c v raman university kota bilaspur (id: u-0084)	18313	0	292	3783
21	dr k n modi university tonk (id: u-0393)	928	56	5292	536
22	dr ram manohar lohia awadh university faizabad (id: u	474414	0	11036	918
23	dr ram manohar lohia awadh university faizabad (id: u	474414	0	11036	787

Fig 3.1.3.4 The final dataset obtained after finding common universities in all three datasets

3. Ranking Algorithm

We have a single file that contains datasets with the parameters like student to faculty ratio, ratio of foreign enrollment to total number of students and the number of scholarships awarded.

According to the different methodologies for ranking the university, the three major university rankings that are Times Higher Education World Rankings, QS World University and Academic Ranking of World Universities assesses colleges or universities on few parameters relating to research, teaching, employability and internationalization.

The performance indicators i.e. the factors that contribute to the ranking process are grouped into five areas in accordance with how much they contribute to the cumulative probability function

- **Teaching**- Student-to-faculty ratio, an indication of commitment to high-quality teaching and support (maximum weightage).
- **Scholarships awarded**- No. of scholarships awarded to students.

- **International outlook-** Number of foreign enrollments. The ability of a university to attract undergraduates, postgraduates and faculty to its success on the world stage.

Computational Equation-

$$\mathbf{Y=ax_1 + bx_2 + cx_3 + d}$$

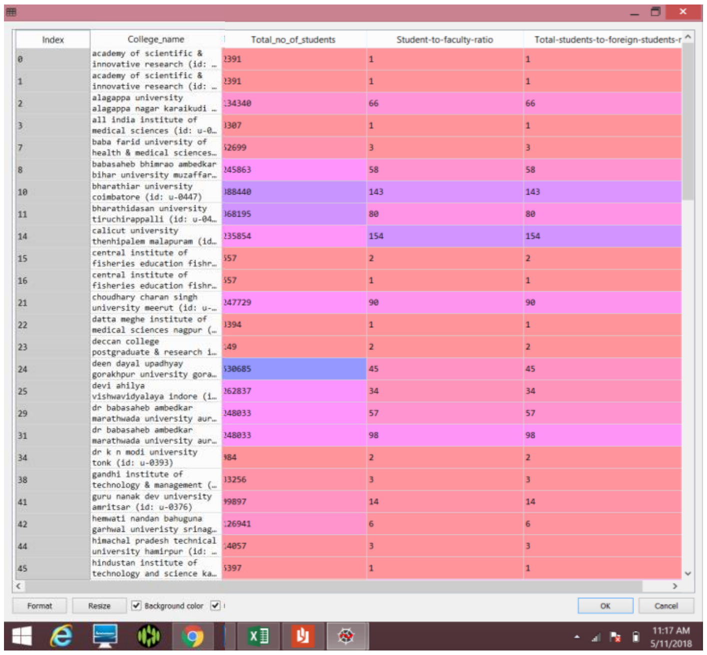
a= weight assigned to student-to-faculty ratio
b= weight assigned to number of scholarships awarded
c= weight assigned to foreign enrollment ratio to total number of students
d= a constant value

x_1= student-to-faculty ratio of a university
x_2= number of scholarships awarded of a university
x_3= foreign student enrollment ratio of a university

<u>Ranking Algorithm</u>

1. Read the input obtained after after cleaning the datasets.
2. Calculate student-to-faculty ratio.
3. Store the value in x_1.
4. Calculate number of foreign enrollments to total number of students.
5. Store the calculated ratio in x_2.
6. Calculate number of scholarships awarded to students.
7. Store the calculated value in x_3.
8. Multiply the values i.e. x_1, x_2, x_3 with corresponding weights i.e a, b, c.
9. Add all the values with weights to get the final value for a particular university.
10. i.e **$Y=ax_1 + bx_2 + cx_3 + d$,** d being some constant.
11. Repeat the steps from 2 till 10 for all the entries of the file.
12. Store the final value obtained by using the equation

13. Apply simple sorting on the column having calculated values.
14. Output the university name with corresponding rankings.



Index	College_name	Total_no_of_students	Student-to-faculty-ratio	Total-students-to-foreign-students-r
0	academy of scientific & innovative research (id: …	2391	1	1
1	academy of scientific & innovative research (id: …	2391	1	1
2	alagappa university alagappa nagar karaikudi …	134340	66	66
3	all india institute of medical sciences (id: u-0…	3307	1	1
7	baba farid university of health & medical sciences…	52699	3	3
8	babasaheb bhimrao ambedkar bihar university muzaffar…	245863	58	58
10	bharathiar university coimbatore (id: u-0447)	388440	143	143
11	bharathidasan university tiruchirappalli (id: u-04…	368195	80	80
14	calicut university thenhipalem malapuram (id…	235854	154	154
15	central institute of fisheries education fishr…	557	2	2
16	central institute of fisheries education fishr…	557	1	1
21	choudhary charan singh university meerut (id: u-…	247729	90	90
22	datta meghe institute of medical sciences nagpur (…	3394	1	1
23	deccan college postgraduate & research i…	149	2	2
24	deen dayal upadhyay gorakhpur university gora…	530685	45	45
25	devi ahilya vishwavidyalaya indore (i…	262837	34	34
29	dr babasaheb ambedkar marathwada university aur…	248033	57	57
31	dr babasaheb ambedkar marathwada university aur…	248033	98	98
34	dr k n modi university tonk (id: u-0393)	984	2	2
38	gandhi institute of technology & management (…	33256	3	3
41	guru nanak dev university amritsar (id: u-0376)	99897	14	14
42	hemwati nandan bahuguna garhwal univeristy srinag…	126941	6	6
44	himachal pradesh technical university hamirpur (id: …	14057	3	3
45	hindustan institute of technology and science ka…	5397	1	1

Fig 3.1.3.5 Output with student-to-faculty and foreign enrollment ratio

CHAPTER 4

PERFORMANCE ANALYSIS

1. Data cleaning

	Before Data Cleaning	After Data Cleaning
Readability	The datasets contained unnecessary columns like College Id, Remarks and punctuation marks.	The dataset is free from unwanted characters and columns.
Better Results	Had ranking algorithm been applied here, the results would have been incomprehensible and illegitimate.	Institution wise appropriate results are obtained after applying the ranking algorithm.

2. Data Processing

Number of rows in Student Enrollment dataset: 656

Number of rows in Scholarships dataset: 20,667

Number of rows in Faculty dataset: 668

Time taken by Python to compare the datasets as per names of the institutions and merge the results into a single file was : 484.2236025333405 seconds

Time complexity of comparison of datasets= $O(n^2)$

3. Ranking Algorithm

Time Complexity of calculating Student-to-faculty ratio = O(n)

Time Complexity of calculating foreign enrollment ratio = O(n)

Time Complexity of sorting the desired values in python= O(n log n)

Overall time complexity of ranking algorithm= O(n log n)

CHAPTER 5

CONCLUSION

5.1 Conclusion

The aim of this project is to produce a list of universities different from the conventional ranking systems that is, on the basis of scope of innovation and sports available in various universities.

In this report, we have studied the need and importance of web crawlers, their various kinds that are in operation today along with their implantation using Scrapy and other tools as well as some of the design issues. We also learned how text analysis can be performed using Python and how it is different from data mining. Also, the final lap of this project, mapper-reducer concept in Hadoop which would be harnessed to handle the big data collected from various websites, was also meticulously studied along with its complexity analysis.

5.2 Future Scope

In this project, we have worked on universities across India only. The scope of this project can be expanded to universities across Asia or further to the entire world. Since this huge amount of data would reach gigabytes, therefore it would be wise to employ Big Data technology. Also, this project can be made more efficient and accessible by offering the information seeker a platform to compare and contrast various universities like we do while buying mobile phones, cars etc. The comparison can be made more accurately by involving other ranking parameters as well for instance, innovation and incubation hubs in an institution, the number of research papers published by it's students and faculty, participation of the institution in various co-curricular activities including cultural and sports fests etc.

<u>REFERENCES</u>

[1] Goodall, Jonathan L., et al. "A first approach to web services for the National Water Information System." *Environmental Modelling & Software* 23.4 (2008): 404-411.

[2] Janbandhu, Rashmi, Prashant Dahiwale, and M. M. Raghuwanshi. "Analysis of web crawling algorithms." *International Journal on Recent and Innovation Trends in Computing and Communication* 2.3 (2014): 488-492.

[3] Castillo, Carlos. "Effective web crawling." *Acm sigir forum*. Vol. 39. No. 1. Acm, 2005.

[4] Leng, Alex Goh Kwang, et al. "PyBot: An algorithm for web crawling." *Nanoscience, Technology and Societal Implications (NSTSI), 2011 International Conference on*. IEEE, 2011.

[5] Narasimhan, Ravi, and T. Bhuvaneshwari. "Big data—a brief study." *Int. J. Sci. Eng. Res* 5.9 (2014): 350-353.

[6] Zende, M. A., Tuplondhe, M. B., Walunj, S. B., & Parulekar, S. V. TEXT MINING USING PYTHON.

[7] Myers, Daniel, and James W. McGuffee. "Choosing scrapy." *Journal of Computing Sciences in Colleges* 31.1 (2015): 83-89.

[8] HEDENBERG, FRED. "Developing components for a data-driven trading system."

[9]A-methodology-for-ranking-universities-and-institutions-in India. *https://www.nirfindia.org/Docs/Ranking%20Framework%20for%20Universities%20and%20Colleges.pdf*

[10] Programcreek. Scrapy Link Extractor. *https://www.programcreek.com/python/example/106165/scrapy.linkextractors.LinkExtractor*

[11] Tutorials Point. Scrapy Spider. *https://www.tutorialspoint.com/scrapy/scrapy_spiders.html*

[12] Stackoverflow. Specific Link Select *https://stackoverflow.com/questions/12145067/scrapy-select-specific-link-based-on-text*

[13] Programcreek. Link Extractor. *https://www.programcreek.com/python/example/106165/scrapy.linkextractors.LinkExtractor*

[14] Webometrics. *http://webometrics.info/en*

[15] Open-Government-Data-Platform. *https://data.gov.in/resources/foreign-students-enrollment-collegses-2015-16*

[16] Analytics-Vidhya Scraping in Python. *https://www.analyticsvidhya.com/blog/2017/07/web-scraping-in-python-using-scrapy/*

[17] Effective Web Crawling: Castillo, Carlos. "Effective web crawling." *Acm sigir forum*. Vol. 39. No. 1. Acm, 2005.

[18] System and Methods for Focused Web Crawling. Chakrabarti, Soumen, Byron Edward Dom, and Martin Henk van den Berg. "System and method for focussed web crawling." U.S. Patent No. 6,418,433. 9 Jul. 2002.

[19] Hand, David J. "Principles of data mining." *Drug safety* 30.7 (2007): 621-622.