Web Content Mining

1 Web content mining

- Data mining is the process of extracting various information from a large collection of databases [5]
- The primary aim for web mining is to extract the useful information and knowledge from the web [5]
  - Finding Web Resources
  - Select type of Information
  - Generalize the Information
  - Analysis the Information
  - Extract Required Information
- Web content mining is the process of collecting the required information from the text, image, audio or videos, already existed on the web. It is also named as web text mining
- Natural Language Processing and Information retrieval are the technologies that are commonly used in web content mining
- Mainly there are two types of approaches for web content mining, namely,
  - Unstructured text mining approach
  - Semi-Structured and Structured mining approach
- Techniques used in unstructured text mining [11, 8, 5]
  - Information Extraction
  - Topic Tracking
  - Summarization

  - Categorization
  - Clustering
  - Information Visualization.

- The techniques used for mining structured data are [11, 8, 5]
  - Web Crawler
  - Wrapper Generation
  - Page content Mining

- Different types of Tools [11, 8, 5]
  - Web info Extractor
  - Mozenda (windows)
  - Screen scrapper
  - Web content extractor
  - Automation anywhere
  - HIT (Hyperlink Image Text Mining)

Tools name	OS	Commercial	What to Extract	Data export format
Automation anywhere	windows	yes	unstructured data or tabular data	Xml, txt, excel, mysql
Easy Web Extract	windows	yes	text, url, image, html	Microsoft Excel, Access, TXT, SQL, HTML, XML, MySQL, ODBC
Web info extractor	Windows 2000/XP/2003	yes	structure or unstructured data	Excel (CSV), Text (TXT)
mozenda	windows	yes	image, text,price, date, address, phone, and fax	CSV, XML, TSV
Screen scraper	windows	yes	Image,text	
Web data extractor	95/98/NT/2000/ Me/XP	yes	Meta tags, urls, text,phone, email address,fax	Microsoft Excel
Web content extractor	windows	yes	text, image, multi media,	Excel, text, HTML, MS Access DB, SQL Script File, MySQL Script File, XML file, HTTP submit form, ODBC Data source
Import.io	Windows,osx,linux	no	text, numbers, locations, URLs, images	CSV, HTML, or XLS.
Web extractor360	Windows 2000/XP/2003/ Vista	no	Images, Phrases, HTML Headers, HTML Tables, URLs (Links), URLs (Keywords), Emails, Phone, Fax	text
Context miner	----	no		CSV, XML
irobotsoft	Windows XP / Seven/ vista/Nt	no	structure or unstructured data	CSV, XML
scrapy	windows, linux , mac, BSD	no	Structure data	CSV, XML,JSON(JavaScript Object Notation)

Figure 1: *Web content mining tools [4]*

Tool Name	Open-source	Platform independent	Record Data	Perform on structured web Data	Perform on Unstructured web Data	Usability	Language	Supported O.S
Web info Extractor	NO	NO	NO	YES	YES	YES	-	Windows
Web Content Extractor	NO	NO	NO	YES	YES	Not for unstructured data	Python	Windows
MOZENDA	NO	NO	NO	YES	YES	YES	Java script	Windows
SCRAPY	YES	YES	YES	YES	NO	YES	Python	Windows Linux, Mac, BSD
SCRAPER SCREEN	NO	NO	NO	YES	YES	YES	Java, Python, Jython NET,VB, ASP,PHP	Windows Linux
Automation Anywhere	NO	NO	YES	YES	YES	YES	Export format XML, Excel TXT, MYSQL,	Windows
RAPID-MINER	YES	NO	NO	YES	YES	YES	Used XML, read ,Excel file	Cross platform
WEKA	YES	YES	YES	YES	YES	YES	Java	Cross platform
ORANGE	YES	YES	YES	YES	NO	NO	Python, C, C++	Cross platform
KNIME	YES	YES	YES	YES	YES	YES	JAVA	Windows Linux

Figure 2: *Comparative study of web content mining tools [10]*

2 SCRAPY

- Scrapy is a web scraping framework [1]
- It s a fully fledged solution which allows people to write small amounts of Python code to create a "spider" - an automated bot which can trawl web pages and scrape them
- Scrapy provides built-in support for selecting and extracting data from HTML/XML sources using extended CSS selectors and XPath expressions [3]
- An interactive shell console for trying out the CSS and XPath expressions to scrape data
- Built-in support for generating feed exports in multiple formats (JSON, CSV, XML) and storing them in multiple backends (FTP, S3, local filesystem)

3 Working with SCRAPY

- Its better to create a virtual environment before installing Scrapy.
- Create the virtual environment with the following command

```
$ python3 -m venv env
$ source env/bin/activate
```

- Virtual environment env (in the screenshot it has named as dj) is created and activated, now install Scrapy as shown in Figure below

```
$ pip3 install scrapy
```

- Now to create a scrapy project enter the following command, Euroncap is the project name which is changeable based on the project

```
$ scrapy startproject Euroncap
```

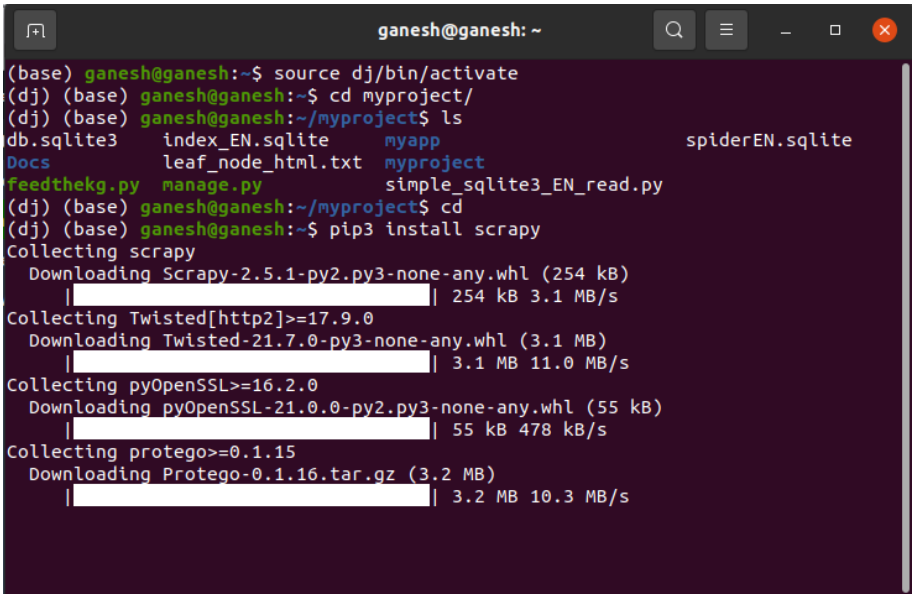

Figure 3: *Installation of Scrapy*

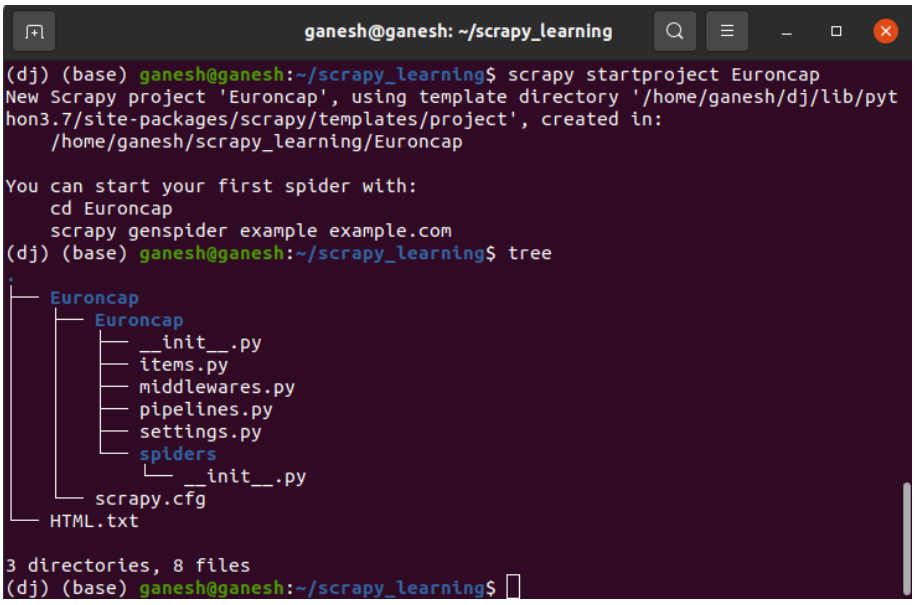

Figure 4: *Tree structure of Scrapy after installation*

- Once the project is created, the tree visualization is presented in Figure 4
- The above files are created to offer different purpose

- **Scrapy.cfg:** This is used to deploy the configuration file
- **Euroncap subfolder:** Project's python module used to import the code from here
- **items.py:** This is used define items/variable definition
- **pipelines.py:** This is used to add pipelines for multi processing like read/downloading files or images
- **settings.py:** This is used to have the settings for scrapy such as how to save the output, where to store the output
- **Spiders subfolder:** This is where the spider code will be placed

- Navigate to spiders subfolder and create a file such as euroncap_spider.py

```
$ cd Euroncap/spiders
$ gedit euroncap_spider.py
```

- After the file has been opened copy the following code

```
import scrapy

class EuroncapSpider(scrapy.Spider):

        # Spider name
        name = "Euroncap"

        # Extracting from local file
        # start_urls = ["file:///home/ganesh/ \
                        scrapy_learning/leaf_node_html.html"]

        # Extracting from website
        start_urls = ["https://www.euroncap.com/ \
                        en/results/toyota/yaris-cross/43819"]



        # Parse functon to extract the features needed
        def parse(self, response):

                # Extracting images from desired location
```

```
                # having data and data-src as per
                # the outline of the page

                # raw_image_urls_data = response.xpath('//*\
                                    [@class="reward-images"] \
                                    //img/@data-src').getall()
                # raw_image_urls = response.xpath('//* \
                                    [@class="reward-images"] \
                                    //img/@src').getall()


                # Extracting all possible images from the links
                raw_image_urls_data = response.xpath('//img \
                                        /@data-src').getall()
                raw_image_urls = response.xpath('//img/ \
                                        @src').getall()


                clean_image_urls = []
                for img_url in raw_image_urls_data:
                        clean_image_urls.append(response.urljoin(img_url))

                for img_url in raw_image_urls:
                        clean_image_urls.append(response.urljoin(img_url))


                #Concentrating only table
                specification_table = response.css("div.tab_container")

                col1_data = [specs.css("span.tcol1::text").getall()\
                               for specs in specification_table]
                col2_data = [specs.css("span.tcol2::text").getall()\
                               for specs in specification_table]


                yield {

               'rating-title' : response.xpath('//div \
                                  [@class="rating-title"]\
                                  /p/text()').getall(),
```

```
                'col1' : col1_data,
                'col2' : col2_data,
                'value'        : response.css('div.value\
                                              ::text').getall(),
                'image_urls' : clean_image_urls

                 }
                        # print("done")
```

- Once the code is copied, save the file.
- Now open the settings.py file and include the following lines below the BOT_NAME definition

```
$ cd ..
$ gedit settings.py
```

```
ITEM_PIPELINES = {'scrapy.pipelines.images.ImagesPipeline': 1}

# Storage folder

IMAGES_STORE = 'Extracted_images'
```

- Now we have written a code to scrape all images from the following webpage "https://www.euroncap.com/en/results/toyota/yaris-cross/43819" to Extracted_images folder under Euroncap subfolder. Along with this the specfication is also scraped and stored in the variables define under yield in the code. This can be saved in json file too.

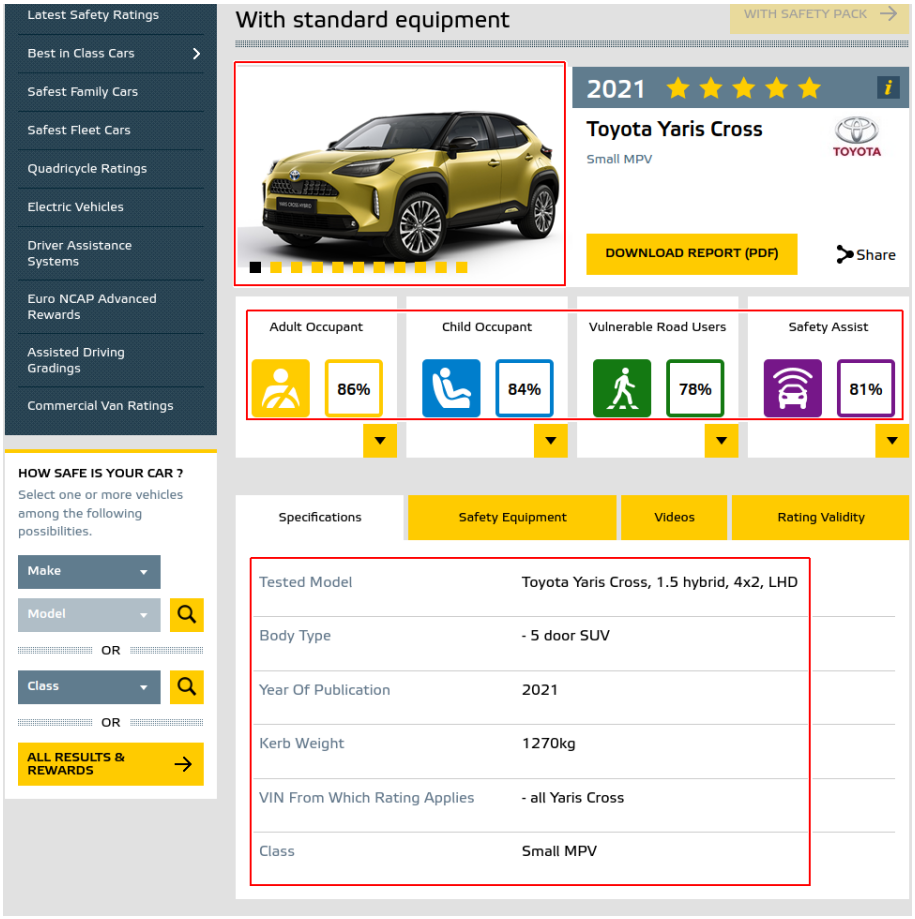

Figure 5: *Info that is to be scraped*

- Now to run the spider enter the following command, Euroncap is the bot name that is provide in the code file for the "name" variable

```
$ scrapy crawl Euroncap
```

- Inorder to store the output in json file enter the following command

```
$ scrapy crawl Euroncap -o euroncap.json
```

- The same procedures can be done for a html file instead of a webpage since the structures are similar.

4 Beautiful soup

- Beautiful Soup is a html parsing package [1]
- It is a helpful utility that allows a programmer to get specific elements out of a webpage (for example, a list of images). As such, BeautifulSoup alone is not enough because you have to actually get the webpage in the first place and this leads people to using something like requests or urllib2 to do that part
- This Python library provides a few simple methods, as well as Pythonic idioms for navigating, searching, and modifying a parse tree [3]
- The library automatically converts incoming and outgoing documents to Unicode and UTF-8, respectively
- This library sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility

Basis	**Beautiful Soup**	**Scrapy crawler**
Structure	It is a library	It is a complete framework
Performance	It is pretty slow to perform a certain task	It can do things quickly because of its built-in feature
Extensibility	It is best for small projects	A better choice for large projects with complexities
Beginner-friendly	It is the best choice for beginners to start with	Scrapy is comparatively more complex than BeautifulSoup
Community	The developer's community of it is comparatively weak	The developer's community of Scrapy is stronger and vast
Consideration	It is considered as a parser	It is considered as a spider

Table 1: Difference between scrapy and beautiful soup [2]

5 Orange

- Open source machine learning and data visualization [6]
- Build data analysis workflows visually, with a large, diverse toolbox
- Add-ons for bioinformatics and text mining [9]
- It can read data from files and process it with ML tools and visualize [7]
- Basically its similar to combining pandas, keras and matplotlib in one application as orange

References

[1] note = Accessed on: 2020-12-22. [Online]. year = 2021 owner = analyticsindiamag url = https://analyticsindiamag.com/scrapy-vs-beautiful-soup-a-comparison-of-web-crawling-tools/ analyticsindiamag, title = Scrapy vs Beautiful soup.

[2] Geeks for Geeks. Scrapy vs beautiful soup, 2021. Accessed on: 2020-12-22. [Online].

[3] note = Accessed on: 2020-12-22. [Online]. year = 2021 owner = hexfox url = https://hexfox.com/p/scrapy-vs-beautifulsoup/ hexfox, title = Scrapy vs Beautiful soup.

[4] Dr. Rozita Jamili oskouei and Zahra Hojati. A comprehensive comparison between web content mining tools: Usages, capabilities and limitations. 08 2015.

[5] T. Suresh Kumar, M. Arthanari, and N. Shanthi. A comparative analysis of different web content mining tools. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 8:1684–1688, 2014.

[6] Orange. Orange data mining, 2021. Accessed on: 2020-12-22. [Online].

[7] Orange. Orange data mining, 2021. Accessed on: 2020-12-22. [Online].

[8] Hari Pandey. Review on web content mining techniques. *International Journal of Computer Applications*, Volume 118:33–36, 05 2015.

[9] predictiveanalyticstoday. Orange data mining, 2021. Accessed on: 2020-12-22. [Online].

[10] Rasha Salman, Mahmood Zaki, and Nadia Shiltag. A studying of web content mining tools. *Al-Qadisiyah Journal Of Pure Science*, 25:1–16, 04 2020.

[11] Anil Sinha, Nidhi Raj, Shameemul Haque, Md Haque, and N.K.Singh. Web content mining: Tool, technique concept. *IOSR Journal of Computer Engineering*, 18:57–60, 12 2016.