Web Scraping To Collect Data From Etl With Pipeline

Saranya.S[1], Preetha.K[2], Shifana Farveen.I[3], Subasri.S[4]
[1]Assistant Professor
[2, 3, 4] Dept of Computer Science and Engineering and Technology
[1, 2, 3, 4] RAAK College of Engineering and Technology,
Pondicherry, Pin-605010,India

Abstract- *Web Scraping is the process of extracting data from web pages, mainly targeting this task are about automated web data extraction.. and finally storing that data into a Csv file. Python language is implemented for carrying out the data from Web pages using requests and BeautifulSoup libraries ETL (Extract, Transform, Load) is a data integration process that combines data from multiple data sources into a single, also responsible for cleaning, their customization and transformation, consistent data store that is loaded into a data warehouse or other target system. It is responsible for the extraction of data, their cleaning, conforming and loading.*

Keywords- Web Scraping; python; data analysis; extraction

I. INTRODUCTION

Data is essentially the plain facts and statistics collected during the operations of a business. It is the raw facts and statistics, specific and organized for a purpose presented within a context that gives relevance and can lead to an increase in understanding and decrease in uncertainty. Web scraping is the method of extracting content and data from the website...it extracts historical data more effectively, of which you can feed such data into some machine learning database. It is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. An ETL pipeline is a set of processes Where raw data is ingested from various data sources and then ported to data store, like a data lake or data warehouse, for analysis. Mechanism in which processes Data pipelines from moving of data from one system with method of data storage processing to another system be stored managed differently.

II. WEB SCRAPING USING PYTHON LIBRARIES

Beautifulsoup

Python library for fetch data out of HTML and XML files. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

Requests

It is one of the integral part of Python for making HTTP requests to a specified URL . It makes a request to a URI, it returns a response and also provides inbuilt functionalities for managing both the request and response.

Pandas

It is used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. It represents the data in a form that is suited for data analysis through its Data Frame.

III. PREVIOUS WORK

Web scraping is the method of extracting content and data from the website...it extracts historical data more effectively, of which you can feed such data into some machine learning database. It is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications.

1. Scraping the data from the webpage:

Scrape match results from the English Premier League for this project. Download the data using a python library called requests, then parse it using beautiful soup to extract what we need, and finally load everything into a pandas data frame so we can clean it up and prepare it for analysis.

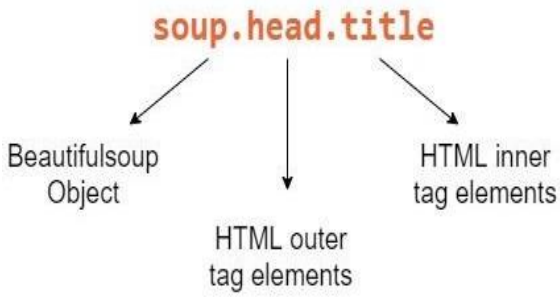
```
In [1]: import requests
In [2]: standing_url = "https://fbref.com/en/comps/9/Premier-League-Stats"
```

The first thing, going to do is figure out how to get the html of a page that shows the EPL standings, and utilize

the requests library to do so, then import it, and to establish the url that going to start scraping.

2.Parsing html links with beautiful soup:

To parse the html, use a library called beautiful soup, which is an excellent library for parsing html. After import the library, the next step is to initialize it with html, call the beautiful soup class and pass in data.txt, which is the html . So now initialized the soup object, to select from the web page.



3.Extract data using pandas and requests:

Use requests to retrieve the value of the html from that address. Typing data. text shows a very long and difficult-to- understand html string, and now need to do is essentially grab this whole table out of the page and turn it into a pandas data frame.

4.Get data with request and pandas:

Grab the data from this page as well, so the first thing find the url of this page from the desired page. In the inspector, we identify all the links and then just keep the one with data which I'll use to scrape the data .

5.Cleaning and merging scraped data with pandas:

Looking at the first five rows using the head technique, the only issue is the multi-level index, which doesn't do much in this situation because the second level of indices aren't particularly useful. With most cases, a multi-level index is not required in pandas. Remove one index level, and there are two index levels because there are multiple rows in bold, implying multiple header rows, and then look at shooting. Here, distinct data frames. That the first row in the data frame has this date and time, and if go back to the matches data frame, they all line up. Then going to use the pandas merge method to join these data frames and apply the result to a variable named team data. Don't want to integrate all of the columns from shooting, however, because many of these columns are simple duplicative, such as right like time comp round venue, and these are duplicated between both data frames. There are taking a few columns from the shooting data frame, and merging on the date column. Looking at team data. head, it was integrated both of our data frames, so I've effectively taken the matches data frame and added.

6.Final results and data frame:

Anchor tags with the class prev, and then take the first one because this returns a list, and then get the href property of that one anchor tag, and then use a format string to convert it to an absolute url. Every time the loop runs, get the prev season's standings url and scrape data for that season, ensuring that we can scrape data from multiple seasons into a single data frame

IV. FUTURE SCOPE

The database engineering is becoming extinct, with data warehousing needs moving to the cloud, and data engineers are increasingly responsible for managing data performance and reliability It is challenging for businesses to exist and stay relevant if they are not good at identifying and adapting to current trends. That is why spotting trends and showing how they change over time is essential and can help companies make suitable decisions. This gives them an edge and helps them stay ahead of their competition. It can easily spot trends with the help of the data gathered during monitoring. Data Analysis

Expressions (DAX) provides a wide range of functionalities for trends analysis. The trending AI Capabilities of Power BI help you visualize the future using predictive analytics and other such big data tools. This can help businesses foresee any need to recruit more employees, change a specific requirement, or further invest in technology.

V. CONCLUSION

Regarding this system, I tried to use and present different semantic web tools and technologies. Although the application itself is not particularly complex, I tried to take care of all the details.I hope I have succeeded in providing a comprehensive view of how the semantic web can increase the power of applications by giving access to more (flexible) data.Further steps may include, for example, an integration with accident data from other nations to create one unique database. In addition, hospital accident records and park attendance data could be connected to the source. These data together would provide a single, comprehensive view that could be used to prevent further accidents.

REFERENCES

[1] Prashant Dahiwale, M. M.Raghuwanshi, and Latesh Malik,"Design and Implementation of Focused Web Crawler Using Genetic Algorithm: An Approach to Web Mining", International Journal of Scientific & Engineering Research, Volume 6, Issue 6, June-2015.

[2] Deepak Kumar Mahto and Lisha Singh "A Dive into Web Scraper World", 2016 International Conference on Computing for Sustainable Global Development (INDIACom).

[3] Emilio Ferrara a,∗ , Pasquale De Meo b , Giacomo Fiumara c and Robert Baumgartner d, "Web Data Extraction, Applications and Techniques: A Survey", Preprint submitted to Knowledge-based systems.

[4] Chain Singh1, Kuldeep Singh2 and hansraj, "A Survey on Web Crawling Algorithms Strategies" , International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue National Conference "IAEISDISE 2014", 12-13September 2014.

[5] Aviral Nigam, "Web Crawling Algorithms", International Journal of Computer Science and Artificial Intelligence Sept. 2014, Vol. 4 Iss. 3, PP. 63-67.

[6] SCM de S Sirisuriya "A Comparative Study on Web Scraping", Proceedings of 8 th International Research Conference, KDU, Published November 2015.

[7] Sergio Flesca, Giuseppe Manco , Elio Masciari and Andrea Tagarelli, "Web wrapper induction: "Abrief survey".