CRAWLING & EXTRACTING STRUCTURED DATA FROM WEBPAGES
DATA SCIENCE – CS838
PROJECT STAGE II

Adarsh Kumar kumar92@wisc.edu

Varun Batra vbatra@wisc.edu

Vibhor Goel vgoel5@wisc.edu

Web Sources:

We chose to extract books and relevant information about them. For the purpose of extraction of data, we chose the below two sources:

Amazon : https://www.amazon.com/
Amazon started as an online bookstore nearly 20 years back. Currently, apart from being the biggest online marketplace of various kinds of goods, it also sells books. It helps you explore Earth's Biggest Bookstore where we can find current paperback books, Kindle eBooks and Audible audiobooks across various genres like Literature, Fiction, Mystery, Thrillers, Cooking, Dating, Comics, Romance, Science, Fantasy etc. They provide details about each book along with the price.

Goodreads : https://www.goodreads.com/
Goodreads is the world's largest site for readers and book recommendations. It helps people find and share books they love. Goodreads was launched in January 2007 and provides extensive details about a book.

Entity Extracted:

We extracted books of different categories like mystery, love, dating, religion, etc. from Amazon. From Goodreads, we extract the list of suggested books everyone should read. The extraction both the sources focuses on most popular and recommended books. This helped us ensure overlaps between entries in the 2 extracted csv tables.

Extraction Methodology and Open Source Tools used:

We made use of the DOM structure of HTML to extract data. We used Python and its library BeautifulSoup to extract the various properties of the books from the HTML DOM structure. Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with a parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It helps in providing a structured way for scraping structured information from web html files. We crawled across multiple pages and extracted book information using the above-mentioned tool. Creating a DOM based crawler for crawling Goodreads was relatively easy as their DOM structure was pretty straight forward. Amazon crawler required some fine tuning of the "class" names to extract the information of use.

For extraction from Amazon, we were able to extract information from a single hit for multiple books. This significantly helped in reducing the number of http requests made. For Goodreads, we first extracted the URLs of most recommended books from their list. We then made a request for each book to extract its data from its URL.

Results:

Number of tuples for Amazon in amazon.csv = 3057
Number of tuples for Goodreads in goodreads.csv = 3001

Schema:

The schema for the 2 tables is described below. As shown, we are focusing on the primary attributes which define a book namely title, author name, rating given by user on two sources, format of the book and the publish date or year. One other entity which we could have used was the number of pages, but this was not available on Amazon with a single request. However, we believe, these attributes are pretty useful for finding the corresponding matches between the two sources for later project stages.

Attribute	Datatype	Description
ID	Integer	Primary Key
Name	String	Book name
Author	String	Writer of the book
Rating	Float	User Rating of the book out of 5 stars
Format	String	The type of book - Kindle/Paperback/Kindle Edition
Year	Integer	The year of book launch