Use case 1 – LinkedIn Job scrapping and report generation

Objective

Develop a web scrapping solution to extract the job listings from the linked portal on a daily basis and generate insights from the scrapped information in form of a report which has to be mailed every day to concerned users.

Fields to be scrapped (This is the minimal requirement that user expects, if some additional parameters can be scraped out for analysis, you are free to use it)

* Job title
* Company name
* Location (optional)
* No. of followers the company profile has
* No. of employees
* Job Type (Contract, Full time etc.)
* Years of experience needed
* Experience level (Intern, Associate, Mid senior etc.)
* Skills needed
* Mode (Remote, Onsite, Hybrid)
* Number of candidates applied
* Easy apply option available or not

**Input**

* List of role/domains to be searched for the day. User should be able to configure this daily as per requirement.

**Output**

* The raw file generated by bot while scrapping information.
* Analytics report providing insights based on scrapped data
  + No. of jobs available for the given input (i.e. “Software Engineer”)
  + Which skills are in demand (A bar chart to visualize)
  + Which mode is dominant (A bar chart or pie chart to show the distribution)
  + Any correlation between no. of employees and no. of applications
  + Any relation between easy apply option available versus the company size (in terms of no. of employees)
  + Analysis of years of experience distribution
  + Any other relationship that can be derived out of the data and can be presented in the daily report.
* The daily report can be in form of excel or PDF where the graphs can either be generated using excel or python frameworks like matplotlib or seaborn.
* Report has to be mailed automatically to concerned user

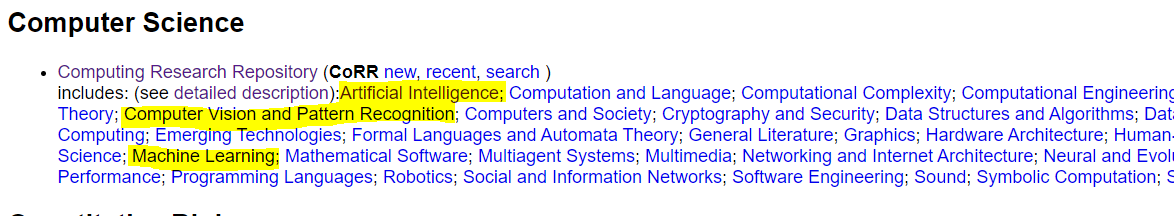
**Note**: The report generation part has to be part of automation pipeline which has to be triggered after the scrapping takes place. For simplicity, you can limit the total no. of jobs to scraped per day to 500 to 1000.

There is no restriction on tools, frameworks to achieve the above tasks.

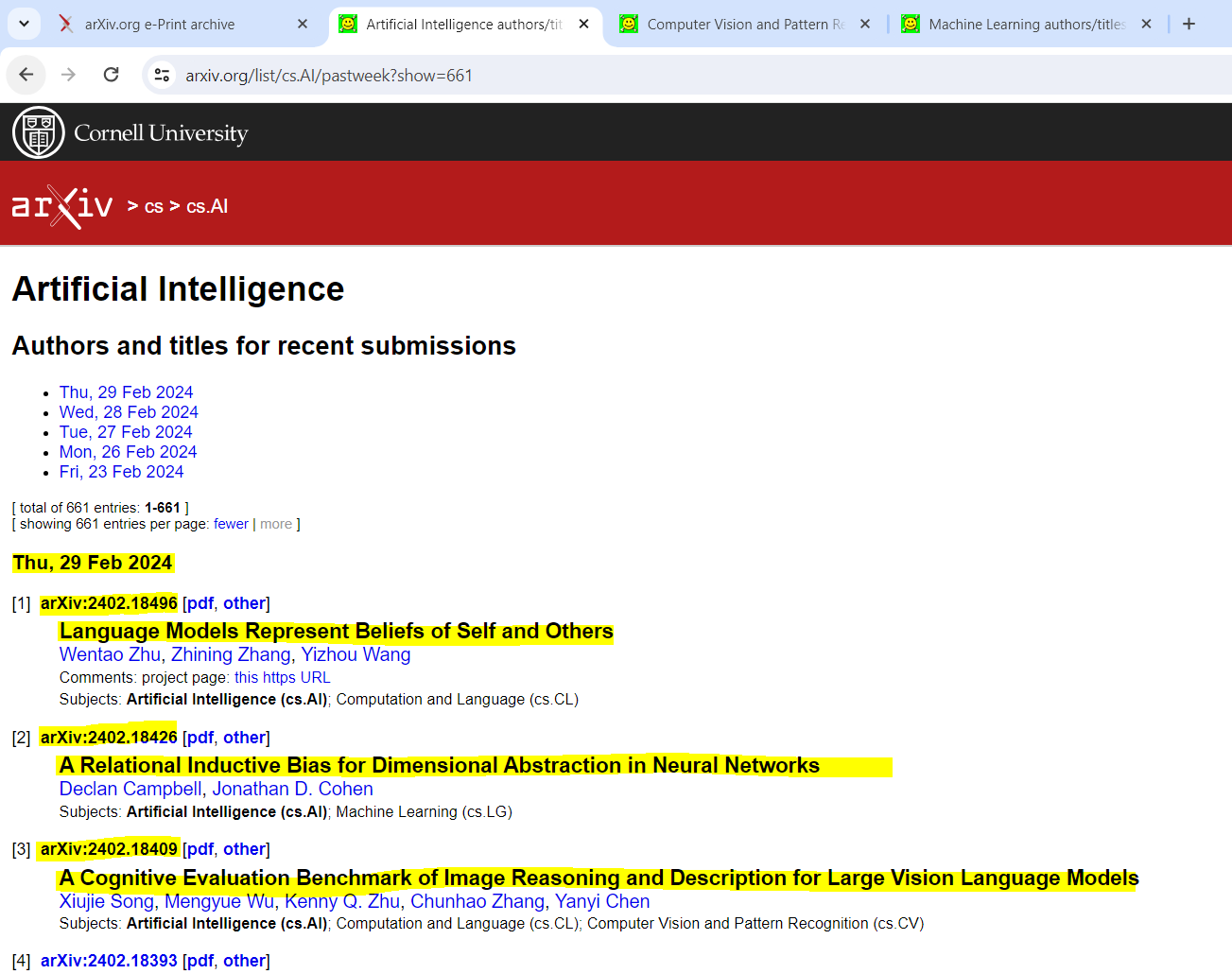
Use case 2 – Arxiv research title scrapping and building inverted index for searching.

Objective

Develop a web scrapping solution to extract the research paper titles along with some more information from arxiv.org for the below marked domains



The bot should move into these subjects and scrape the title and ID (for checking if the paper title is already downloaded or not) along with the link for pdf. Bot has to make sure that the date is matching with today’s date and also select “Show All” as initially, the results will be paginated.



After scrapping the bot has to build an inverted index. More details will be provided in a separate document explaining how to construct and infer results using this data structure.

Use case 3 – Yahoo finance report generation

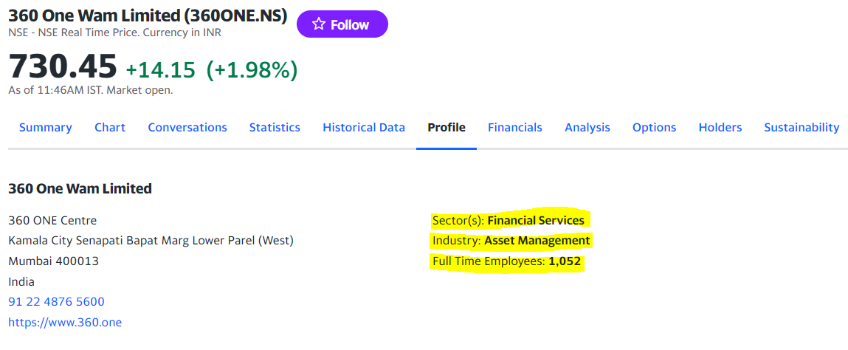
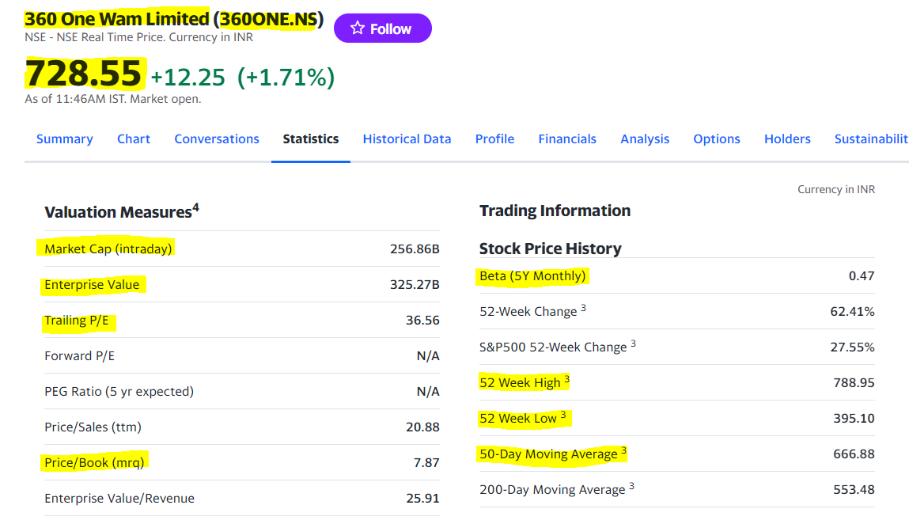
# Objective

From the list of top 500 stocks, scrape the details for each of these companies and generate a comparison report of stocks. The expectation is a pipeline which will scrape the details, generate the comparison report and mail to concerned user.

**Input**

List of Top 500 shares (Nifty 500) in excel sheet

**Output**

* Get the following details from the yahoo finance
* Given a templated excel file, generated the report
* Send it through email to concerned recipients

**Note**: The excel sheet will be attached along with the document. Please refer the excel sheet for more details