ASSIGNMENT -1

**In Assignment 1, I was presented with a challenging JSON file that includes the data representing DMCA notices and created a data pipeline to translate and investigate the data using Python and R.**

**How I did it**

1. **Pre-analysis and knowledge Pre-analysis and knowledge**

* First, I validated and formatted the JSON into file with JSONLint to get to know more about its nested structure.
* The next thing that I did was to format a clean JSON file that can be easily inspected and parsed without writing any code.

1. **Flatten the JSON**

* In Python, and R I wrote a code to loop through reads of each notice, each work, and each infringing URL within works.
* I constructed a row of each separate URL with metadata of notice ID, sender name, principal name, and description of work.
* Saved the flattened data as CSV (flattened\_step1.csv) to ensure to check them.

1. **Add two columns, domain, IP address (parallelized)**

* In Python, I used urllib.parse and in R I used urltools package to get the domain of each of the URLs.
* Applied a parallel step of IP resolution:
  + In Python: ThreadPoolExecutor with 4 work threads, socket.gethostbyname ().
  + R: with a 4-core cluster: nslookup via parSapply().
  + Mapped the resolved IP addresses back to the DataFrame and stored a new CSV (flattened\_step3.csv).

1. **significances by creating summarizations**

* **Summary 1**: The 10 most infringing URLs domains.
* **Summary 2**: count of unique notices by sender.
* **Summary 3**: Infringing URLs by work description group.

Moreover, such additional numeric indicators were computed as:

* URL leader in the senders.
* Leading work and domain by URLs.
* Observation of the highest amount of URLs.
* Unique number of domains in the top sender.

Exported summaries into individual CSV files.

1. **Tools and technological stack**

* **Technologies** Languages: Python and R.
* **py libraries**: pandas, json, urllib.parse, socket, concurrent.futures.
* **R packages**: jsonlite, dplyr, urltools, parallel.
* **Methods**: JSON-processing, DataFrame manipulation, parallel processing, domain parsing, IP address resolution, summarization.

1. **Outcome**

* Effectively designed a scalable data pipeline to flatten, enrich and summarize the JSON data.
* Provided CSV outputs and summary stats.
* Submitted high quality and clean code, worked with Python and R with good analysis and data engineering capabilities.