**Infringement URL Analysis Summary**

**Objective:** The goal of this project was to analyze a dataset containing nested JSON structures with infringing URLs. The analysis involved flattening the data, extracting and processing URLs to identify their domains and IP addresses, and generating insightful summaries to reveal trends and patterns related to online infringement activities.

**Steps Taken:**

1. **Data Flattening:**
   * Utilized Python’s json\_normalize to flatten nested JSON data.
   * Extracted relevant fields such as infringing URLs, IDs, types, titles, and dates.
   * Saved the flattened data into a CSV file for further analysis.

python

Copy code

df = json\_normalize(data['notices'], record\_path=['works', 'infringing\_urls'],

meta=['id', 'type', 'title', 'date\_sent', 'date\_received',

['works', 'description']],

record\_prefix='infringing\_')

df = df.rename(columns={'infringing\_url': 'infringing URL', 'works.description': 'work\_description'})

df.to\_csv('flattened\_json\_data.csv', index=False)

1. **Domain and IP Address Extraction:**
   * Implemented functions in both Python and R to extract domains from URLs and resolve them to IP addresses.
   * Python functions used urlparse and socket libraries, while R functions leveraged URL parsing and DNS resolution methods.
   * Applied parallel processing using Python's ThreadPoolExecutor and R’s future\_map for efficient handling of large datasets.

python

Copy code

with ThreadPoolExecutor(max\_workers=4) as executor:

df['IP address'] = list(executor.map(get\_ip, df['domain']))

r

Copy code

df <- df %>%

mutate(Domain = future\_map\_chr(infringing.URL, extract\_domain, .progress = TRUE),

IP\_Address = future\_map\_chr(Domain, get\_ip, .progress = TRUE))

1. **Handling Missing Values:**
   * Removed rows with missing domain values due to their minimal impact on the dataset.
   * Imputed missing IP addresses with "Not Available" to maintain data integrity.
2. **Data Summarization and Analysis:**
   * Created visualizations and summaries to understand the data:
     + **Count of Infringing URLs per Domain:** Identified the most frequently used domains for infringing content.
     + **Top 10 Most Frequent Infringing URLs:** Found that Google Drive links were predominant.
     + **Top 10 Most Frequent IP Addresses:** Highlighted IP addresses associated with infringement.
     + **Most Pirated Works:** Analyzed works most frequently targeted by infringers.
     + **Fake URLs Received Per Day:** Visualized trends in potentially fraudulent URL reports over time.

**Results:**

* The analysis revealed that Google Drive is a major platform for sharing pirated content.
* Certain IP addresses and domains were identified as hotspots for infringement.
* Trends in fraudulent activities were mapped over time, providing insights into spikes or patterns.

**Performance Considerations:**

* The script efficiently handles large datasets through parallel processing, utilizing up to 4 CPUs. This ensures timely and effective analysis of extensive data.

**Future Work:**

* Potential improvements could include expanding the dataset to include more sources or refining the analysis to detect new patterns in infringement.

**Contributing:**

* Contributions and feature requests are encouraged. Please refer to the issues page for details on how to contribute.

**License:**

* The project is licensed under the MIT License.