# Problem set – Data Scientist Position – Forward Analytics

## Task 1

For Task 1, I downloaded the Level 1 Data LEI Records from the GLEIF website to begin parsing the LEI data. To understand the structure, I referred to the LEI-CDF v3.1 XML schema, focusing on fields of interest: Entity_Name, LEI, Legal_Code, and Legal_Form. Parsing the XML required defining a namespace, specifically {'lei': 'http://www.gleif.org/data/schema/leidata/2016'}. Once loaded, I converted these fields into a pandas DataFrame for better data manipulation.

Since the Legal_Form field was only available as a code, I needed to convert it to text. For this, I used the ISO 20275 legal forms code list, but to manage memory limitations, I selectively loaded only the "Entity Legal Form name Transliterated name (per ISO 01-140-10)" and "ELF Code" columns. After merging this data, I obtained a DataFrame with Entity_Name, LEI, and Legal_Form. However, industry and company size data were still missing.

To map company names to industries, I searched for relevant datasets and identified a free dataset from People Data Labs. Due to memory constraints, I loaded only limited columns, specifically 'name,' 'industry,' and 'size,' and then merged it with the LEI data. This unfiltered data contained 35,425 entries, many of which lacked industry or legal form information. After filtering, I was left with 386 entries. The results are available in the out folder.

## Task 2

For Task 2, I focused on cleaning company suffixes and deduplication. First, I dropped suffixes, a challenging process since I could not find a comprehensive list of all possible suffixes. I used a data-driven approach, extracting suffixes from the entries and saving them in a file. I then leveraged GPT to identify common industry suffixes, which I used to create a regular expression that removed suffixes, resulting in some duplicate names. I subsequently dropped entries with duplicate names, reducing the dataset size from 23,446 to 23,043.

Addressing different spelling variations was the next challenge, for which I explored common text similarity measures such as Levenshtein distance, fuzzy matching, and cosine similarity. Comparing each entry to all others proved too time-consuming and inefficient, so I limited comparisons to entries sharing the same prefix. For Levenshtein distance, I observed a reduction from 23,446 to 23,414 entries within 115.8 seconds using a threshold of 2, and similarly, from 23,446 to 23,414 with a threshold of 85. Cosine similarity yielded a reduction from 23,446 to 23,425 within 200.5 seconds with a threshold of 0.7, although this approach encountered an out-of-vocabulary error due to prefixes such as "E" in "E J O."

| Method | Prev size | Updated |
|---|---|---|
| **Levenshtein** | 23446 | 23414 |
| **fuzzy matching** | 23446 | 23420 |
| **cosine similarity** | 23446 | 23425 |