Automated Job Scraper for Canadian Banking & Insurance Companies

Ana Carrera

Description

This Project builds an automated web scraper that collects job postings from major Canadian banking and insurance companies. The scraper extracts data-related roles (e.g., Data Analyst, Data Scientist, ML Intern) and consolidates them into a single CSV file. The goal is to reduce the time spent manually checking multiple career portals and create a unified dataset for job exploration.

Programming Environment

Python 3
Libraries: requests, BeautifulSoup (bs4), pandas
Environment: VS Code / Jupyter Notebook / Terminal

Goals / Objectives

Primary Objective

To automate the process of collecting data-related job postings from Canadian financial institutions into one clean, searchable dataset.

Specific Objectives

- Scrape job postings from 8-10 major Canadian employers
- Extract relevant fields (title, company, location, link, posting date)
- Filter for analytics/ML/data roles using keyword matching
- Save all results into a standardized CSV file
- Provide a basic interface for quick review
- Build a foundation that can be expanded into NLP or automation later

Dataset Description

This project generates its own dataset by scraping job postings from publicly available career portals on major Canadian institutions.

The final dataset is created dynamically each time the scraper runs, and it contains structured information about data-related job postings across the Canadian banking and insurance sectors.

Company Selection Criteria

Canadian banking and insurance companies were chosen using three criteria:

1. Industry Relevance

   These companies are among the largest employers of Data Analysts, Data Scientists, Machine Learning Engineers, and Business Intelligence roles in Canada. Companies with tons of data. This aligns directly with the goal of preparing for Toronto's job market.

2. Publicly Accessible Job Pages

   Every selected company provides:
   I. Publicly accessible job listings
   II. No login or authentication requirements
   III. Predictable, stable HTML or JSON structures

   This makes them scrape-friendly and compliant with ethical scraping guidelines.

3. Toronto-Centric Hiring

   All selected organizations hire heavily in the Greater Toronto Area, where data roles are concentrated.

Specific Career Pages Used for Scraping

Banking Sector (Big Five Banks):

1. RBC
2. TD
3. Scotiabank
4. CIBC
5. BMO

Insurance Sector (Major National Insurers):

6. Manulife
7. Sun Life
8. Aviva
9. Intact
10. Desjardins

Building the Scraper Function

The scraper function automates the process of collecting job postings from publicly available career pages. Instead of manually checking each company's website, the function visits a job page programmatically, downloads the HTML behind it, and uses BeautifulSoup to parse and search through the page. This allows us to extract specific information such as job titles, links, and locations.

Because every company formats its job listings differently, the function is written to be fully configurable through input parameters. These include:

- url: the career page to scrape.
- company and sector: metadata added to the final output for labeling, not for scraping logic itself.
- title_selector, link_selector, location_selector: CSS selectors that tell BeautifulSoup where to find the titles, hyperlinks, and locations on that specific website. This makes the scraper reusable across companies with completely different HTML structures.
- base_url: used to correct relative URLs that appear in many job portals.

Once the function locates the job listings, it loops through them and extracts the data into a consistent dictionary format. Each job posting includes fields like title, company, sector, location, and URL. The function then returns a list of these job dictionaries, which can later be combined into a Pandas DataFrame and saved as a CSV.

In short, the scraper function converts messy website HTML into clean, structured job data that you can analyze, filter, and reuse across the entire project.

Building the Main Function

The main script coordinates the overall workflow. It begins by defining a list of companies to scrape, then loops through each one and calls the scrape_jobs() function to extract their job postings. All results are collected into a single list, converted into a Pandas DataFrame, and saved as a CSV file. In this way, the main script acts as the controller of the project: it specifies which career pages to visit, triggers the scraping process, and produces one unified dataset that can be used for analysis.

Testing Stage

The script was ran using only RBC and TD