Group Project

Python For Data Analysis

Motivation

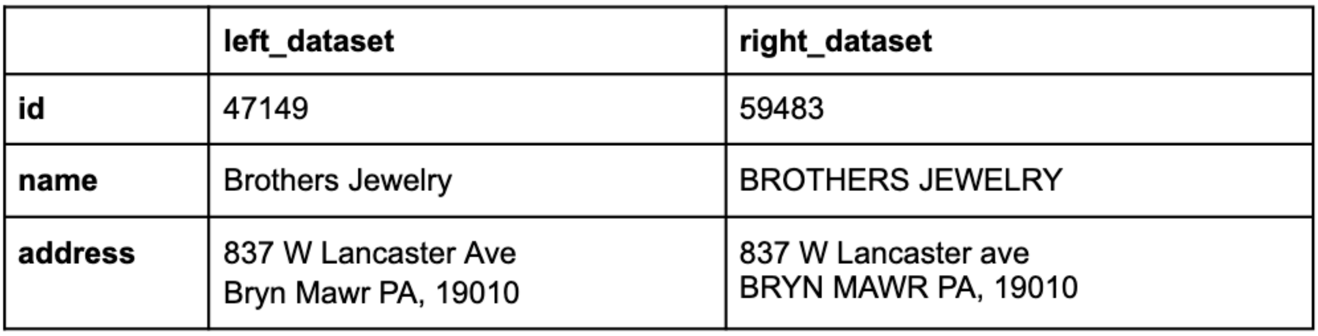
You work as a Data Analyst for a tech company that sells data about small to medium businesses in the US. A prospective client provided your organization a list of businesses. You are tasked with determining if the businesses are available on your data platform.

Problem Statement

You are given two datasets:

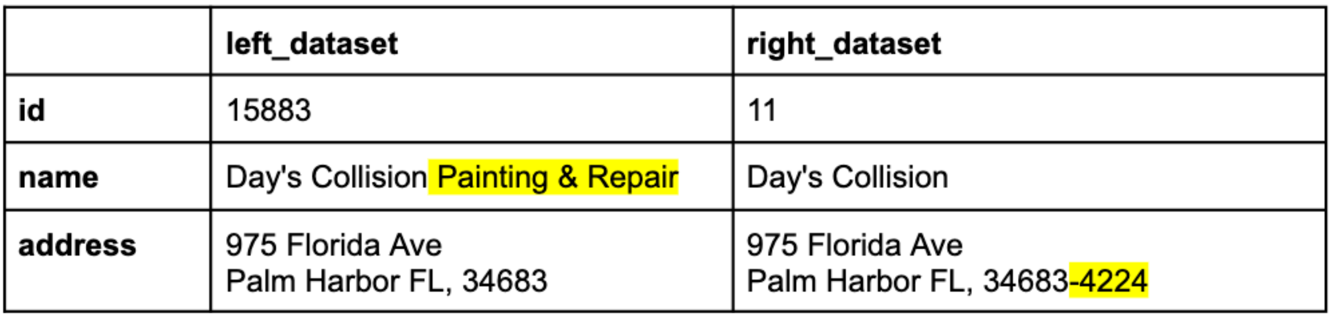
- left_dataset.csv
- right_dataset.csv These datasets contain business names and their addresses.

The goal of this project is to find the businesses that are common to both datasets, that is, the businesses that have a name and address that match between the left and right datasets. Here is an example of a match. It is a nearly perfect match, ignoring case, since the name and address have identical values:



	left_dataset	**right_dataset**
id	47149	59483
name	Brothers Jewelry	BROTHERS JEWELRY
address	837 W Lancaster Ave Bryn Mawr PA, 19010	837 W Lancaster ave BRYN MAWR PA, 19010

The `left_dataset.csv` and `right_dataset.csv` come from different sources so the same business can have subtle differences between datasets. For instance, in the example below, the names and zip codes do not match exactly, but these entities clearly point to the same business:



	left_dataset	**right_dataset**
id	15883	11
name	Day's Collision Painting & Repair	Day's Collision
address	975 Florida Ave Palm Harbor FL, 34683	975 Florida Ave Palm Harbor FL, 34683-4224

Here are a few more examples of businesses that match but have slight differences across the left and right datasets:

	left_dataset	**right_dataset**
id	15925	2206
name	Jazz House Supper Club	Jazz House Supper Club LLC
address	9331 E Adamo Dr Tampa FL, 33619	9331 East Adamo Drive Tampa FL, 33619

	left_dataset	**right_dataset**
id	89855	72
name	Esposito's Italian	Esposito's 1948
address	14306 N Dale Mabry Hwy Tampa FL, 33618	14306 N Dale Mabry Hwy Ste F Tampa FL, 33618-2052

In order to maximize the number of matches that you find, you will probably need to use sophisticated methods that can find close matches in addition to identical matches. When your algorithm runs, it should produce a list of triplets consisting of the entity_id from the left dataset, the business_id from the right dataset, and a confidence score. The confidence score should have values between 0.0 and 1.0 and convey a sense of confidence of the match. An identical match should have a score of 1.0.

Your submission should consist of matches that have a high degree of confidence, eg greater than 0.8.

Here is a sample submission (as a csv file) for the examples shown above, where the confidence scores are just examples and do not come from an actual calculation.

```
left_dataset, right_dataset, confidence_score
47149, 59483, 1.0
15883, 11, 0.99
15925, 2206, 0.95
89855, 72, 0.91
```

Deliverables

Your submission will consist of a single zip-file. The zip file will be subdivided into sections that contain the data, any exploratory work, your source code, your presentation slides in pdf format, a Jupyter Notebook that generates the final results and a requirements.txt file.

More specifically, the structure of the zip file should be organized as follows:

```
{project_name}/
    data/
        left_dataset.csv
        right_dataset.csv
    explorations/
        {notebook1}.ipynb
        {notebook2}.ipynb
        ...
    presentation/
        Project - Spring 2024.pdf
        {team_slides}.pdf
    src/
        {module1}.py
        {module2}.py
        ...
    requirements.txt
    results.ipynb
```

The `results.ipynb` file should contain minimal code that kicks off your calculations and displays the results in the notebook. All of the underlying functions (and/or classes) should be implemented in the modules in the `src` directory.

The `src` directory should contain all of the functions (and/or classes).

The `explorations` folder can contain any number of Jupyter Notebooks that you used for exploratory work.

The `presentation` folder should contain this PDF file and a copy of the team slides (PDF) that you present in your talk.

The `requirements.txt` file should contain all of the packages that are imported within your code base. There are no limitations on the packages you import, you are free to import any packages you would like.

Responsibilities

Each team member should evaluate one python package for the project. If your team has 6 team members, your project should have 6 different result sets from 6 different python packages. For the final result, you can pick the best performing package or take some combination of the top performing packages.

Recorded Presentation

The presentation should be no longer than 6 minutes. Each team member should present their contribution to the project. You don't need to spend any time introducing the project, you can dive straight into your results.