Data Engineering Project

OVERVIEW:
Welcome! Your job here is to take the data sets below and produce a entity resolution solution for the USE CASE (below).
The basic evaluative principle is how well you do navigating, understanding, and engineering a solution for matching on two unseen data sets that contain real world data with independent issues.

In evaluating your solution, we prioritize:

- Code cleanliness and testing. To this end, please take the time to refactor your code to make it understandable to someone that has not worked on this problem before.
- Solution design and strategy. Our team builds systems that scale as well as possible, and are as general as possible in order to promote re-use. In practice, this means that we make minimal assumptions about a problem beyond what is defined in a spec, and write very general, object-oriented solutions. Please make sure to include in your write-up an explanation of your strategy, with justifications for your choices, and additional thoughts on how you would refactor or improve your solution given more time.

USE CASE:
A political data vendor has provided us with ~1M voting profiles which we would like to match up against ~160K of the resume data vendor's profiles in a similar region for the purposes of machine learning predictions and graph construction. We have filtered down these two data sets to a reasonable set of features for you to match upon. You must clean and transform these rows and provide what you would deduce as exact identity pairings.

DELIVERABLES:
- **exact_matches.csv** consisting of columns political_id, resume_id representing unique entity resolution pairings (i.e. rows from each data file you have determined represent information for the same individual in the real world). You may include additional "feature" columns which represent how some one was matched (e.g. a boolean with 0/1 called "same_first_name").

- **readme.pdf** a write up detailing your methods, procedures and evaluations. In particular, it should include the following metrics:
  - Exact Matches Percentages (relative to the original datasets)
  - Ambiguous Dataset Entities Percentages (same)
  - Machine Runtimes (e.g. on AWS instance types)

  Explain how you are defining an exact match, ambiguous dataset entities *and why these definitions are reasonable.* Furthermore, this document should also have your thoughts on what you could do with this data if you worked with it daily (as opposed to this brief time).
- **match.py** all the code (Python *highly* preferred, R, Scala etc) you used in this assignment. Using what you have given us and the source data, we must be able substantively to replicate your results.

GROUND RULES:

- It is encouraged that you do not work on this project for more than 8 hours and should be delivered 3 days after the send date via email with attachments or appropriate google drive links.
- You may not have any person help you. You may, however, consult the Internet, your books, your notes, your parrot, etc.
- The appropriateness and quality of your approach is an important factor in our evaluation (though we understand you had limited time). Provide evidence you have wrestled with the data and pertinent algorithms to squeeze out what knowledge you can in the time you have.
- You must write up your results and submit all the code you used to build your solution.

DATA SETS:

political_data_vendor.csv - This file contains a small subset of columns useful for entity resolution from an unprocessed regional database from a political data vendor.

Field names:

- political_id
- first_name
- last_name

- city
- birth_year
- gender

resume_data_vendor.csv - This file contains a small subset of columns useful for entity resolution from an unprocessed regional database from a professional resume vendor.

Field names:

- resume_id
- first_name
- last_name
- degree
- degree_start
- local_region

SUGGESTIONS:

- Spend 30 minutes exploring and understanding the data.
- Set up a matching strategy plan and spend 60 minutes data cleaning (e.g. are there bad birth dates and how will we handle these?), 2-3 hours performing the matching and generating results, then the remaining time writing up your methods and refactoring the code.
- Standardizing columns without losing match potential is critical.
- Creating or sourcing alias tables is allowed.
- A minimal approach should achieve at least 30% of all the resume data rows paired to a unique political data row, as a baseline for reasonable matching definitions.