Radial Analytics

Skilled Nursing Facility Coding Exercise

Project: The Centers for Medicare and Medicaid Services (CMS) releases *Nursing Home Compare* datasets for Skilled Nursing Facilities (SNFs) on a monthly cycle. Included in these datasets are facility administrative information, quality information, and deficiencies & penalties information for roughly 15k SNFs.

The aim of this project is to create a simple command-line search interface into the *Nursing Home Compare* dataset that will allow a user to generate a list of facilities by searching by geographic and other facility characteristics.

Data: The most recent set of Nursing Home Compare data is freely available for download at:
  https://data.medicare.gov/data/nursing-home-compare
Click on the "Download CSV Flat Files (Revised) Now" link to download a complete set of up-to-date CSV files.

Specification: Please write, document, and share code in Python or the language of your choice that meets the following design criteria:

Execution:
- The code can be run from the terminal/shell of an Ubuntu or OSX machine.
- Inputs can are specified via command-line arguments (e.g., see the argparse python parser).
- Outputs are printed directly to the console in JSON format, where they can be redirected directly to a .json file.
- Code execution instructions are documented.

Inputs:
- The following input arguments can be passed to the interface:
  - **zip_code** – required. The five-digit zip code that geographically anchors the search.
  - **search_radius** – search radius in miles (default: 10). The radius around the zip code centroid within which to conduct the search.
  - **min_overall_rating** – optional (default: 1, range: 1-5). The minimum allowable overall quality rating for each returned SNF. The overall_rating may be found in the "ProviderInfo_Download.csv" file.

Outputs:
- Outputs should be returned as a properly-formatted human-readable JSON list of SNF mappings. The keys for each SNF mapping should be as follows:
  - **name** – SNF name from the "ProviderInfo_Download.csv" file.

- o **address** - SNF address from the "ProviderInfo_Download.csv" file.
- o **city** - SNF city from the "ProviderInfo_Download.csv" file.
- o **state** - SNF state from the "ProviderInfo_Download.csv" file.
- o **zip_code** - SNF zip code from the "ProviderInfo_Download.csv" file.
- o **phone** - SNF phone number from the "ProviderInfo_Download.csv" file.
- o **overall_rating** - SNF rating from the "ProviderInfo_Download.csv" file.
- o **lat** – latitude of the SNF zip code centroid
- o **lng** – longitude of the SNF zip code centroid
- o **distance_miles** – distance in miles from the centroid of the search zip code centroid to the SNF zip code centroid
- o **score (optional)** – the SNF score analytic, as described in the following "Score Analytic" section.
- The output list should be sorted in ascending order by the **distance_miles** metric. Facilities with identical distance_miles values should be ordered in descending order by **overall_rating**.

Score Analytic (optional):
- Develop a **score** analytic that provides a goodness of fit score for each SNF based on geographic proximity and SNF quality, taking the following factors into consideration:
  - o Distance from the geographic anchor.
  - o SNF overall rating. Caveat: this rating is recognized within the clinical community for being not highly correlated to true SNF quality. While of some value, you may want to down-weight it in favor of hard metrics.
  - o Any other relevant metrics you identify within the available data that you deem appropriate.
- Be creative, and document the motivation and justification for your score analytic formula and variable choice. Explain and justify the chosen range for your analytic.

Geographic search:
- To simplify distance calculations between the geographic anchor and each SNF, we are including a "zip_code_centroids.csv.gz" file that reports the centroid lat and lng for all zip codes in the United States. To calculate the distance in miles between the input zip_code and each SNF, simply calculate the distance between the input zip code and SNF zip code centroids- there is no need to geocode any SNF addresses.
- To implement a distance calculation between two (lat, lng) pairs, you will find the haversine formula to be quite useful.

Style:
- For optimal readability, the submission should meet all PEP 8 style guidelines for Python code.

Code Submission: Please submit your documented code to Eric, including any special instructions to install dependencies and to execute the code. As part of your submission, please discuss the efficiency of your code, as well as any considerations for productizing and scaling the code.