Assignment (Role: Machine Learning Engineer)

As you may know, it is very important to us to have the best possible data to define the businesses we, at Reputation, are analyzing. We pull data from various sources, by various means and at various times. It doesn't always fit together well.

This is an exercise to get a sense for how you might approach building a scalable and maintainable data quality enhancing and monitoring system which can fit together "messy" data.

Attached is a set of "messy" sample dataset about US Auto dealerships.

This data includes:

- Location ID - A distinct ID for each dealer location
- Dealership Name
- A Tenant ID - An ID representing parent organization of dealer
- Address - Location detail attributes (address_line, city, state, zip, latitude, longitude)
- Phone Number
- Website
- Google url - google listing of the location
- FB url - FB listing of the location
- Cars url - Cars listing of the location

This data is gathered from various sources and is of varying levels of quality. Not all values are filled in for every row and not all of the data in here aligns perfectly. There are a lot of duplicate entries, overlapping entries, mismatched entries, entries with wrong/outdated values and missing entries. None of these values can be trusted fully.

Task is to approach developing a scalable and maintainable system/pipeline which can help monitoring and enhancing this data. On a high level, system should output the following

- Best estimate of a "canonical" list of distinct dealerships in this dataset
  - Along with (as best as system can figure out) the name, address, phone number, and website of those dealerships
  - As well as the google, facebook, and cars.com websites for those dealerships if system can determine them
- System should be able to map every one of the entries in the original dataset to that canonical list, along with a confidence score of how confidently system is concluding that this is the same dealership
- System should be able to handle new dealership data coming in future and scale
- System should generate following reports
- Total number of canonical locations
- Total number of data issues found
- Lists of potential correction (add/update) of any fields for each record in the original dataset.

One way to format this would be to create a csv with columns for

    - Location ID
    - Attribute Name (e.g. name, phone number, google url)
    - Current value
    - New value
- Data quality by tenant

One way to format this would be to create a csv with columns for

    - Location ID
    - Tenant ID
    - Data Quality %

The expectation here is

- Not that you build full working system
- But clearly demonstrate how you would approach this and showcase the first pass at this
- Clearly lay out the development plan and how you would proceed further
- Possibly prepare a notebook to demonstrate your idea/algorithm to the core problem (messy data)
- Explain any 2 methods that you would prefer to push the project to production for client usage
- A document for the overall development plan (modules/stages, APIs, interfaces, deployment strategy)
- Brownie points if you could build a RESTful webservice with the algorithm using any framework (Flask/Django) that you are convenient with. (High Level Deliverable)

Hints:

None of the attribute values can be trusted. We may need a probabilistic approach.