Spark! Heyrick Research Fuzzy Matching Deliverable 1

Team: Suzy Kirch, Yang Hu, Chengyang He, Haoran Kang

Part 1: All data has been gathered

- Only had to gather Yelp data (using Yelp fusion API) for validation.
- Since Yelp fusion API has limited 5000 calls, it took 3 days to get all the data.
- Gathered for requested states of MA and MO.
- Data stored as json files and converted to csv files for further exploration. All data files were organized by states and zip codes.
- Most data are complete. After meeting with clients and managers, we decided to fill the blank with NaN for any missing data.
- We standardized the phone number for each business by E164 format, which keeps country code for each number.
- Since Yelp data's address contains geographical location as well, we dropped the longitude and latitude from address and made new attributes for them in the dataframe.

Part 2: Initial results

Because our project is a matching problem, we don't have any small questions to start with. This is the process we decided to move forward with.

1. We will begin with address matching. The address match will be the primary match, as no business can "fake" its address.
    a. We will split the address, first matching state, then zip code. Only after we have that split, will we work on street address matching.
    b. Some zip codes are xxxxx - xxxx. We can, for all intents and purposes, ignore what occurs after the dash, as that is extra information for the post office, so a match on the first 5 is sufficient.
2. After getting an address match, we will compare name and phone numbers as checks.

In regards to algorithms, we will be looking at spaCy rule matching. Within spaCy, there is a rule-based entity recognition that could be helpful, especially because we are primarily matching strings.