**Amenity data capture approach**

The solution needs different kinds of amenities available close to a neighborhood, to perform an effective clustering computation as it tries to find matching neighborhoods for the user’s current neighborhood.

**Data sources**

Multiple data sources such as the Google Places API, Yelp API, Foursquare API, OpenStreetMaps and datasets on multiple websites were evaluated.

Google Places, Yelp and OpenStreetMaps along with public datasets we found were dropped due to a mix of reasons including lack of a healthy dataset, difficulty to capture data and cost of API usage.

Foursquare API was chosen as the target amenity data source as it provided the following benefits for us;

* Availability of multiple types of amenity data (‘venues’ in Foursquare lingo) due to it being a location based platform for places.
* A high free data rate limit allowing us to make 5,000 API calls per hour and a maximum of 99,500 APi calls per day.
* Availability of latitude and longitude with each venue which allows is to map them to the census tract data that form the foundation of our data set.
* The ability to efficiently filter location types (grocery stores, schools, restaurants etc).
* Tehe ability to search by zip code.

The zip code list was obtained from <https://simplemaps.com/data/us-zips>.

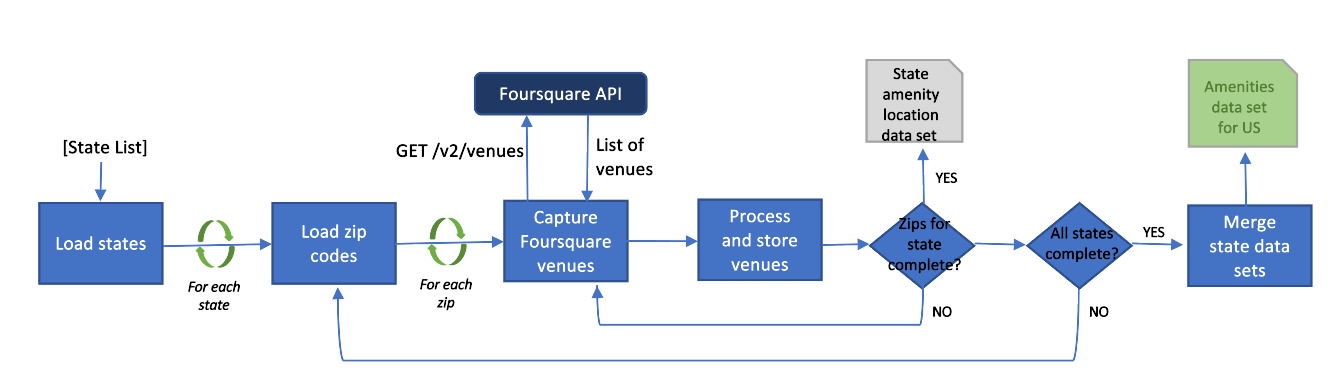
**Data acquisition**

A python data capture script was created which will start with a list of state codes covering continental United States. For each state it will filter the list of zip codes from the US Zip Code list that is inputted. Then for each zip code an API call is made to the Foursquare /venues endpoint with the parameters intent, client ID, client secret, near, category ID and version.

Ex: <https://api.foursquare.com/v2/venues/search?intent=browse&categoryId=4bf58dd8d48988d1e7941735&client_id=<CLIENT_ID>&client_secret=<CLIENT_SECRET>&v=20201020&near=43240>

The response is a JSON response with a list of ‘venues’. Each venue JSON object is processed to capture name, address, postal code, latitude and longitude. Each of these records get added to the amenity data set for a certain category such as ‘grocery\_stores’.

An amenity list CSV file is created for each state. Once all states’ data are received, all get consolidated to a single CSV file.



**Data preparation**

While the dataset from the Foursquare API is comprehensive, we did notice that the data quality had some issues such as the same location being duplicated, multiple names for the same kind of venue and incorrect type tagging. These were specific data quality issues to clean up through manual means rather than an automatic approach.

The consolidated CSV file which has data across all states for a type of amenity (ex: grocery stores) was loaded to Open Refine for data cleaning. Following cleaning steps were done.

* Convert altitude and longitude to an equal length.
* Run multiple rounds of clustering to find similar names of locations and merge them.
* Cluster address data and look for duplications.
* Create multiple text facets for the name and look for suspicious names that may not match the amenity type we are interested in and investigate if these are accurate entities to retain.