Data Engineering Career Track
Open-ended Capstone Step 2: Project Proposal

Problem Statement:
Create an API that leverages NYC 311 Service Request data to allow potential renters and buyers to search the volume and type of requests for a given location and radius.

Motivation:
The pace and sheer volume of the New York City residential real estate market can make buying or leasing property tricky. In addition to the overall condition and any amenities that might be available, the setting of a living space is critically important and is difficult to assess. Shrewd renters or buyers might try to visit the area around their potential new home at different times of day or get in contact with people living nearby to screen for nuisances, but the process is fraught with error.

311 is the non-emergency phone number for the city government. From here, callers can access resources and report all kinds of issues including noise pollution, vermin and pests, parking violations and utility outages. In recent years, the city also accepts some 311 reports through a smartphone app. Investigating the reports around a potential home could save buyers and renters significant heartbreak. This project aims to create an API that could be used alongside listings to help solve this problem.

Interface:
The final product of this project should accept latitude, longitude, and a radius and return a summary of the 311 reports that pertain to that area in JSON.

Phases of development:
Initial work on this project will focus on exploratory analysis, to discover which of the many complaint types, descriptors, location types are relevant to this use case. Complaints about hired drivers or homeless encampments within the subway shouldn't impact a location's summary as reported by the API.

Following this, an initial ETL process on the CSV provided by NYC OpenData will need to be developed. The CSV should be read into a temporary table within a suitable RDBMS, where superfluous columns and rows that don't meet the criteria established in the first phase can be dropped before being inserted into a final table.

Next an ETL process that queries the Socrata Open Data API (SODA) of the 311 database on a daily basis will need to be collected. Fortunately getting access to the last day's data is as simple as constructing an HTTPS to the endpoint with the following get parameters, where YESTERDAY and TODAY should stand in for the appropriate date in ISO8601 Times format:

- limit=50000
- where=created_date between '{YESTERDAY}' and '{TODAY}'

The resulting JSON will need to be parsed, again being filtered as with the CSV before being inserted into the production table.

Finally, the software to handle client requests, running the queries against the production table, and responding with the summary will need to be developed.

Possible Technologies:

- **PostgreSQL**
- **Python 3**
- **MVC or API platform**
    - **Flask**
    - **Django**
- **Scheduler / Orchestrator**
    - **cron**
    - **Airflow**

Scaling into the cloud:
Being limited in scope, the above API could be implemented as a serverless application (using for example AWS's API Gateway, AWS lambda and AWS RDS).

Dataset:
New York City publishes *311 Service Requests from 2010 to Present* as a CSV file to NYC OpenData on a daily basis and through a Socrata Open Data API (SODA). As of April 2022 this dataset contained 28.3 million rows, was 15 gigabytes in size and includes report details such as complaint type, responding city government agency, latitude and longitude.

Resources:

- **311 Service Requests - NYC OpenData**
- **311 Service Requests - SODA**