



Where to live in SF?

A Renting Guide for NEU Students



Presentor: Xinyi Feng

Course: CS5001

Professor: Mark Miller

Date: May 2nd 2022

Where to live in SF?

01

Inspirations

Why did I choose the topic?

02

Conceptions

What was I planning to do?

03

Tools

What tools I used?

04

Solving & Coding

How did I code?

05

Future improvements

What will be improved?

06

Conclusion

Reference and links

01

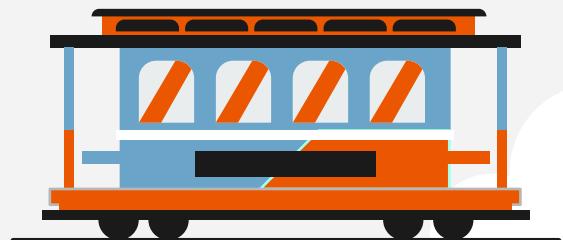


Inspirations

Why did I choose the topic?

Inspirations :

- Having renting & housing issues myself
- Love to explore cities
- Devoted to helping students
 - No customized renting websites for NEU students



02



Conceptions

What was I planning to do?

When you search a place to live in SF

A brainstorming time



Is it a house or apartment?
Is it a studio or 1b1b or...
Is a new house?

...



How much I need to pay?
Any cheaper choices?



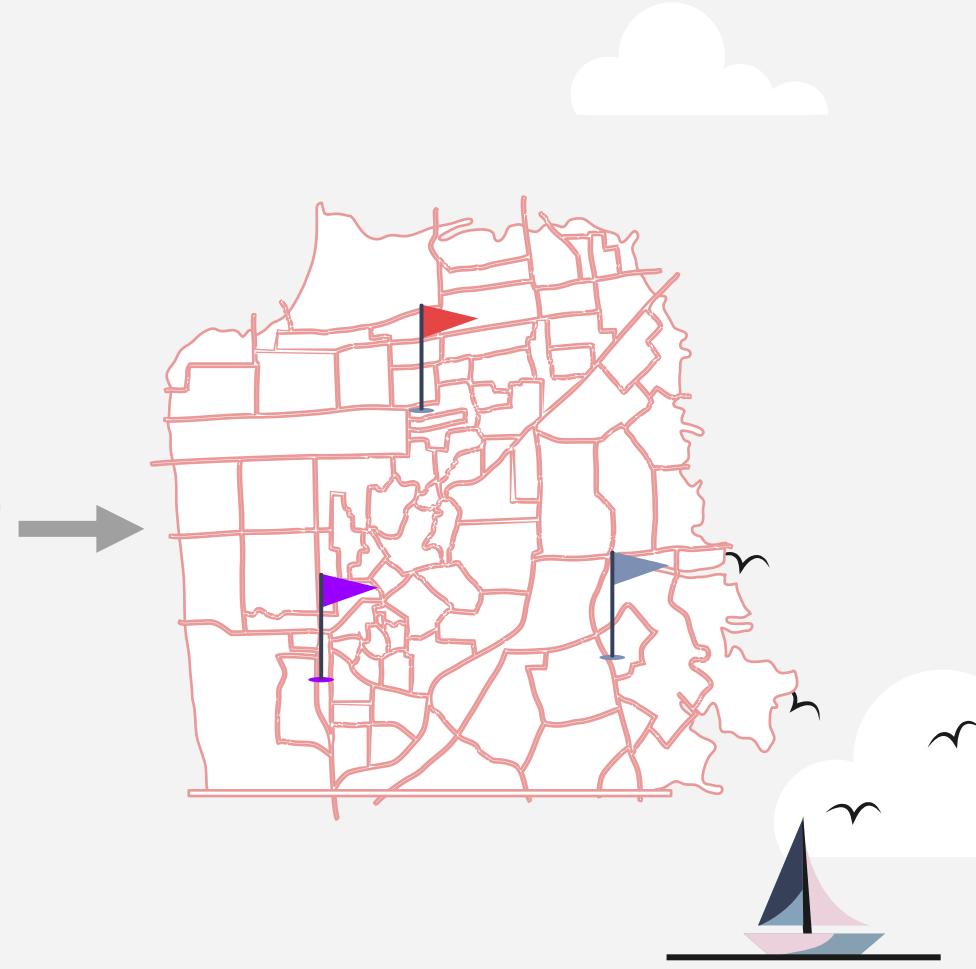
Is the neighborhood safe?



How far it is from school?
How can I get to school?



Focus :



03



Tools

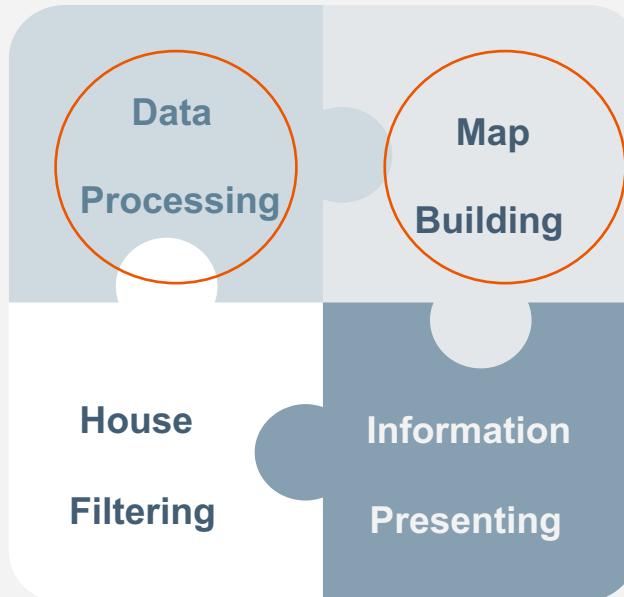
Why tools I use? And why?

Tools :

Pandas – Library to analyze data

Geopy - Client to get zip code

Google Distance Matrix Api
– Client to calculate
commute time



Folium – Library to draw map

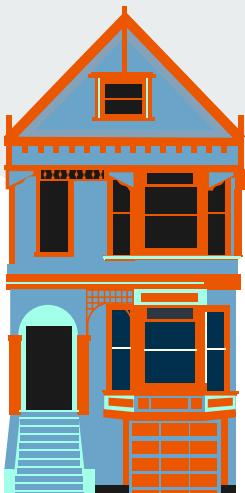
Mapbox – Map base

Jupyter Notebook – Server

Python, Html – Language use



04



Coding

Full codes is in *crime_data_process.py*
And *map_building_house_adding.py*

See the full version in jupyter notebook:
final project_full version.ipynb

See the complete html work:
<https://fentpam.github.io/WhereToLiveInSF/>

Coding : Pre -part—Processing data of Crime rate:

Full codes is in crime_data_process.py

Source from:
<https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783/data>

Incident Date	Incident Time	Incident Day of Week	Incident Category	Intersection	Analysis Neighborhood	Latitude	Longitude	Point
10/1/2021	19:45	Friday	Assault	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
5/17/2021	14:00	Monday	Burglary	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
5/27/2021	2:00	Thursday	Burglary	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
6/8/2021	7:00	Tuesday	Motor Vehicle Theft	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
10/20/2021	10:00	Wednesday	Motor Vehicle Theft?	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
10/20/2021	10:00	Wednesday	Stolen Property	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
10/20/2021	10:00	Wednesday	Motor Vehicle Theft	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
10/20/2021	10:00	Wednesday	Motor Vehicle Theft	AVENUE N \ 04TH TI ST	Treasure Island	37.82357442	-122.3637428	POINT (-122.36374276695297 37.82357442446163)
6/2	import pandas as pd import geopy							

```

def get_zipcode(df, geolocator, lat_field, lon_field):
    """ Converts latitude and longitude into zipcodes
    :parameters:
        df: a pandas dataframne, with two columns longitude and latitude
        a Geopy library use OpenStreetMap nominatim
        https://geopy.readthedocs.io/en/stable/#geopy-is-not-a-service
        lat_field: name of the latitude column
        lon_field: name of the longitude column
    :returns:
        A number with 5 digit (zipcode)
        -1 when there is value err or key err raised
    """
    try:
        location = geolocator.reverse((df[lat_field], df[lon_field]))
        print(location)
        return location.raw['address']['postcode'][5:]
    except (ValueError, KeyError):
        return -1

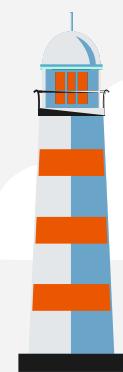
def main():
    # reads data from original csv file named "crime.csv"
    data = pd.read_csv("crime.csv")
    # utilizes data in latitude and longitude columns
    df = data[['Latitude', 'Longitude']]
    geolocator = geopy.Nominatim(user_agent='xinyi_sf_map')
    # processes the first 100K rows
    zipcodes = df.head(100000).apply(
        get_zipcode,
        axis=1,
        geolocator=geolocator,
        lat_field='Latitude',
        lon_field='Longitude').value_counts().rename_axis('ZipCode').to_frame('Count')
    # writes into a new csv file for further use with two columns: zipcode and crime count
    zipcodes.to_csv('crime_rate_output.csv')

```

More than 500,000 lines data

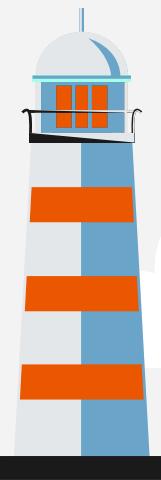
Two columns data sorted by zip code

Zipcode	Crime count
94103	8462
94102	6778
94133	6044
90103	5035
94143	3946
94134	3810
94124	3686
94164	3469
94104	3319
94131	3313
94112	2924



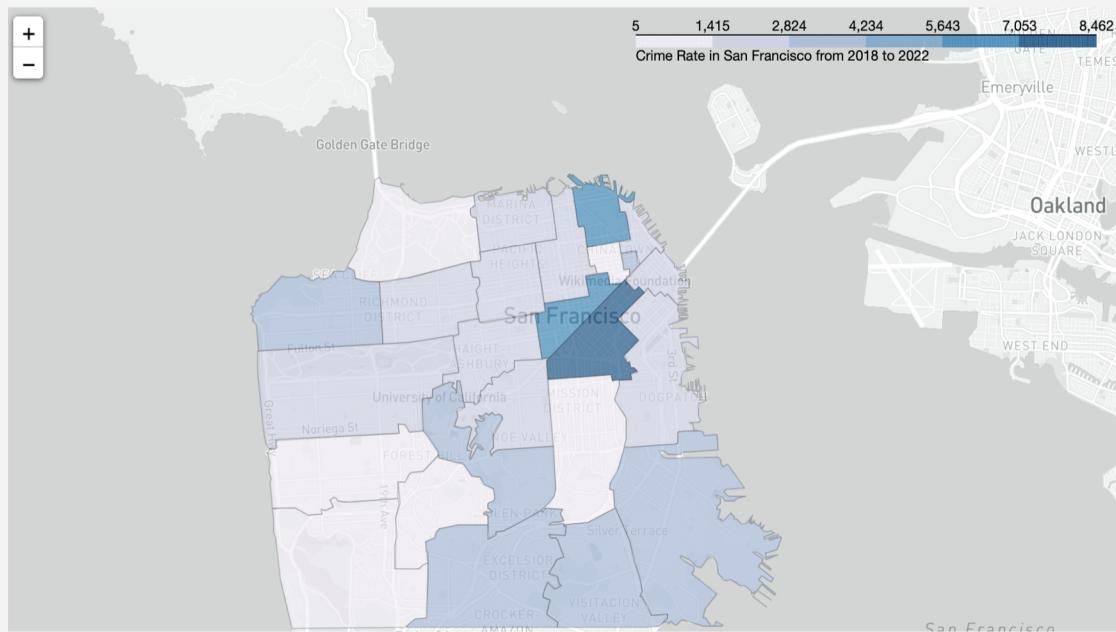
Coding : Main -part--Building Maps with multi layers

SF-Base Map:



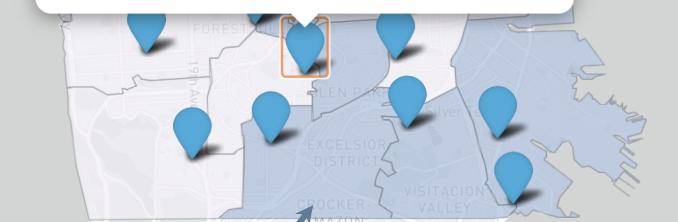
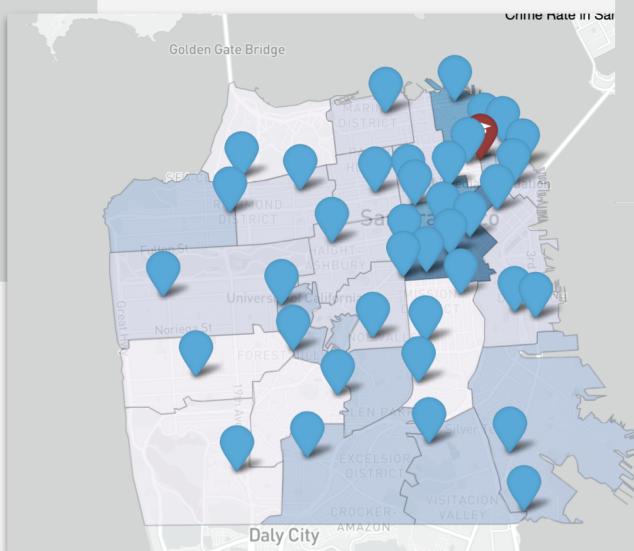
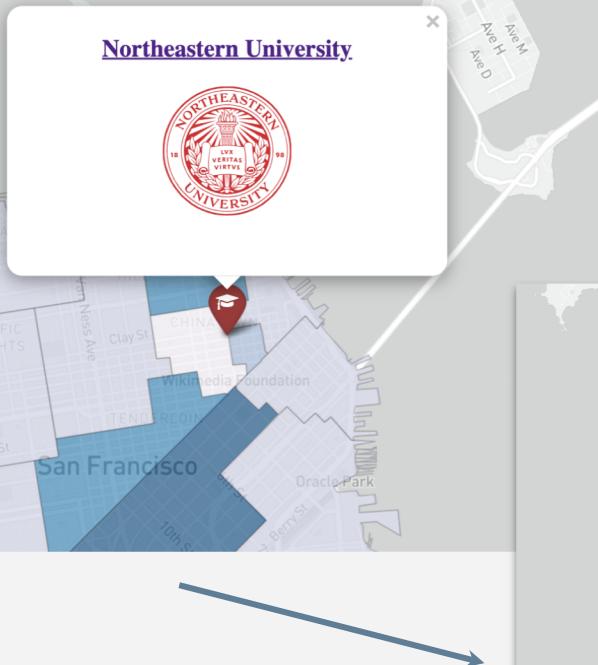
Coding : Main -part--Building Maps with multi layers

Crime rate layer map:

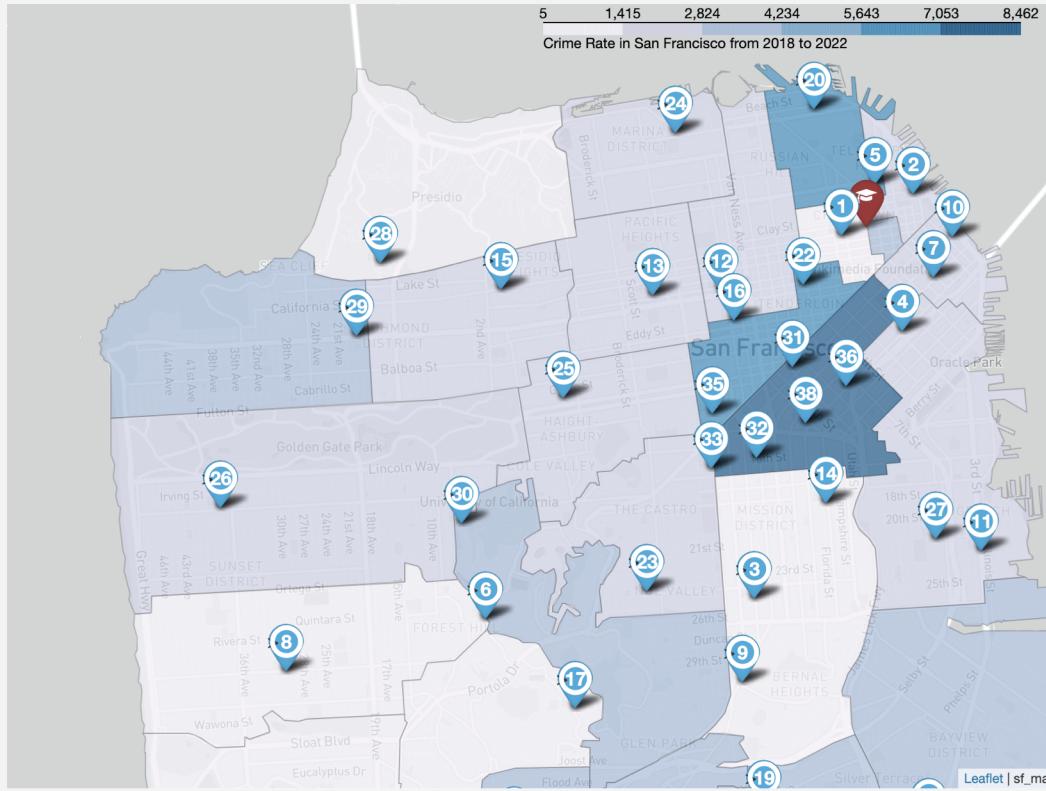


Coding : Main -part--Building Maps with multi layers

NEU and House layer map

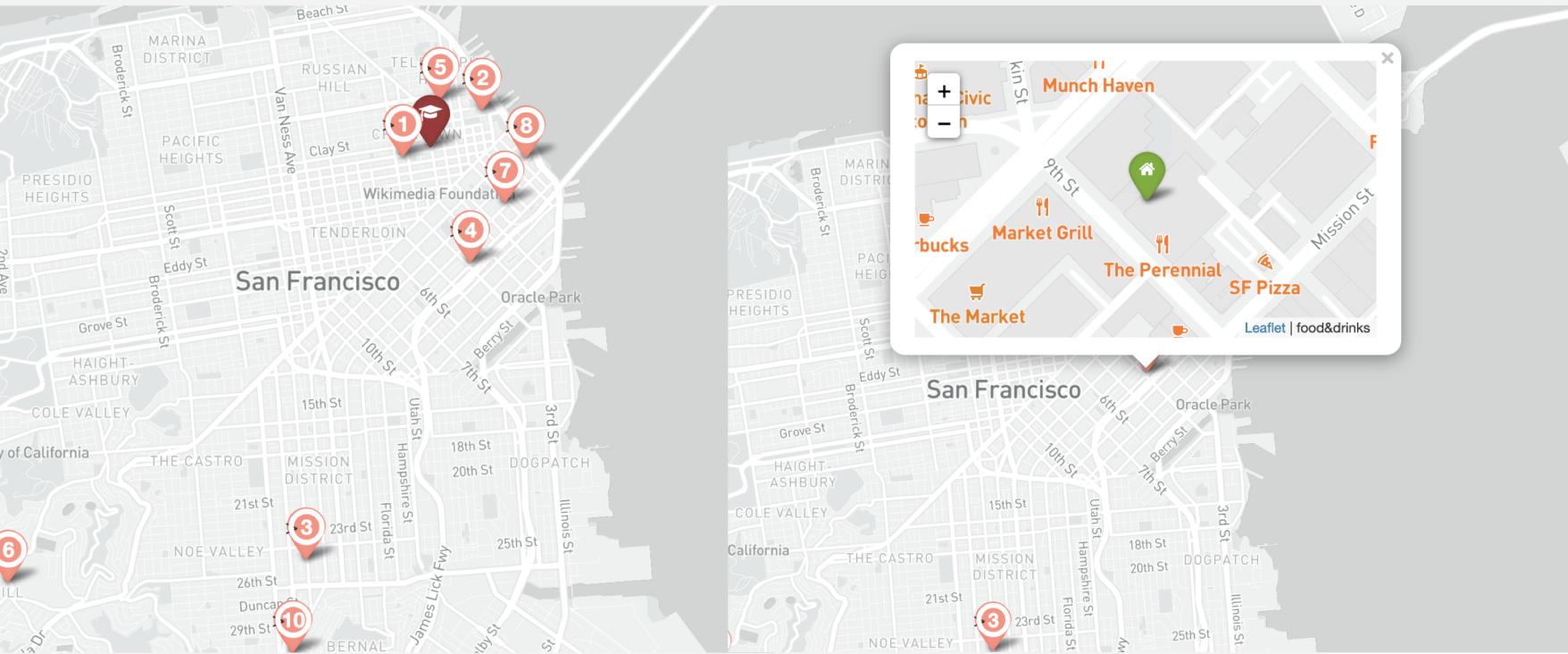


Coding : Main -part--Rank houses by simulated algorithm
Ranks house on the map



Coding : Main -part--Selected top 10 houses with nearby info

Top 10 house with nearby info provided



05

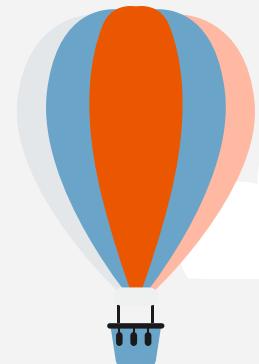


Future Improvements

Always seeking for better

Future Improvements

- Updated and comprehensive housing source :
Python crawler
- Algorithms improvement:
To be Precise
To be User-interactive
- Features to be added:
Nearby facilities adding
Detail commute routes showing
Job searching (companies and opening positions)
Forum for NEU neighborhood(finding roomies...)
... more to work on



06



Conclusion

You gotta to think!

IN PROCESS

SOLVED

House source

Data Processing

Crime rate

Interactive Interface

Map Building

Geo Info of SF

Advanced Algorithm

Calculating & Filtering

Map Base with Layers

Detailed & Customized Features

Recommend House

Simulated Algorithm

House Presenting with Partial Info

Presenting On The Map

Reference with links :



[Click to see my whole project](#)

- **Data source:**

[Police Department Incident Reports: 2018 to Present | DataSF | City and County of San Francisco.](https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783/data)(n.d.). Retrieved April 15, 2022 from : <https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783/data>

[zillow.](https://www.zillow.com/) (n.d.). Retrieved April 17, 2022 from: <https://www.zillow.com/>

[mapjson.](https://geodata.lib.berkeley.edu/catalog/ark28722-s7jg7w)(n.d.). Retrieved April 17, 2022 from: <https://geodata.lib.berkeley.edu/catalog/ark28722-s7jg7w>

- **Python Libraries:**

Pandas Documentation.(2022, April 06). Retrieved April 16, 2022 from: <https://pandas.pydata.org/pandas-docs/stable/>

Folium 0.1.5.(2015, Aug 13). Retrieved April 16, 2022 from: <https://pypi.org/project/folium/0.1.5/>

- **Server:**

[Jupyter Notebook](https://docs.jupyter.org/en/latest/index.html) (n.d.). Retrieved from: <https://docs.jupyter.org/en/latest/index.html>

- **Client:**

[Google Distance Matrix API.](https://developers.google.com/maps/documentation/distance-matrix/overview)(n.d.). Retrieved from: <https://developers.google.com/maps/documentation/distance-matrix/overview>

[Geopy](https://geopy.readthedocs.io/en/stable/)(n.d.). Retrieved from: <https://geopy.readthedocs.io/en/stable/>

- **Mapbox studio** (n.d.). Retrieved from: <https://www.mapbox.com/mapbox-studio/>



Thank you !

Do you have any questions?



www.linkedin.com/in/feng-xinyi



feng.xinyi@northeastern.edu



<https://github.com/FentPam>