ARCTIC Summer Camp 2024:

Predictive Modeling of House Prices Using Crime and Education Metrics

By Rinisha Ramprakash and Nethmee Perera

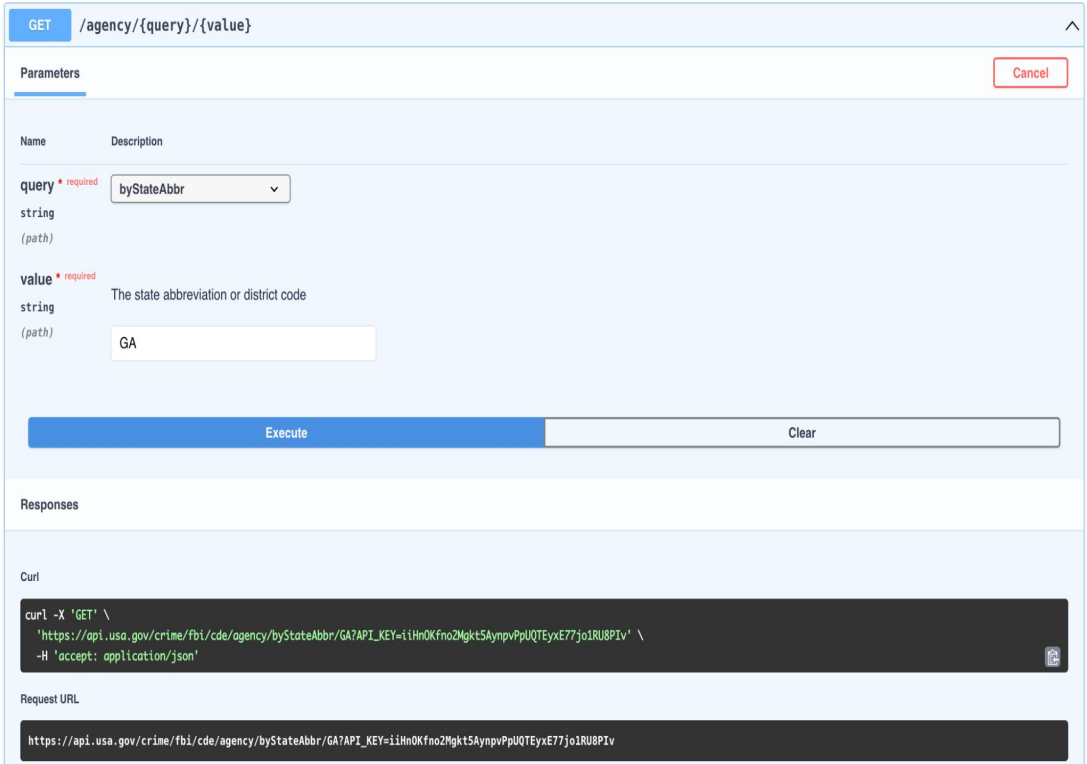
Introduction: This project aims to predict house prices by analyzing crime and education data.

Objective: To develop a predictive model that estimates house prices based on the surrounding crime rates and education quality

Scope: Project covers data collection, data preprocessing, model development, evaluation and data visualization

Crime Data Collection from FBI API

```python
import requests

def get_data_from_uri(uri):
    # Set up the headers with the API key
    headers = {
    "Content-Type": "application/json"
    }

    # Send a GET request to the API endpoint
    response = requests.get(uri, headers=headers)

    # Check the status code of the response
    if response.status_code == 200:
        data = response.json()
        return data

    else:
        print(f"Request failed with status code {response.status_code}")
        print(response.text)
        return None
```

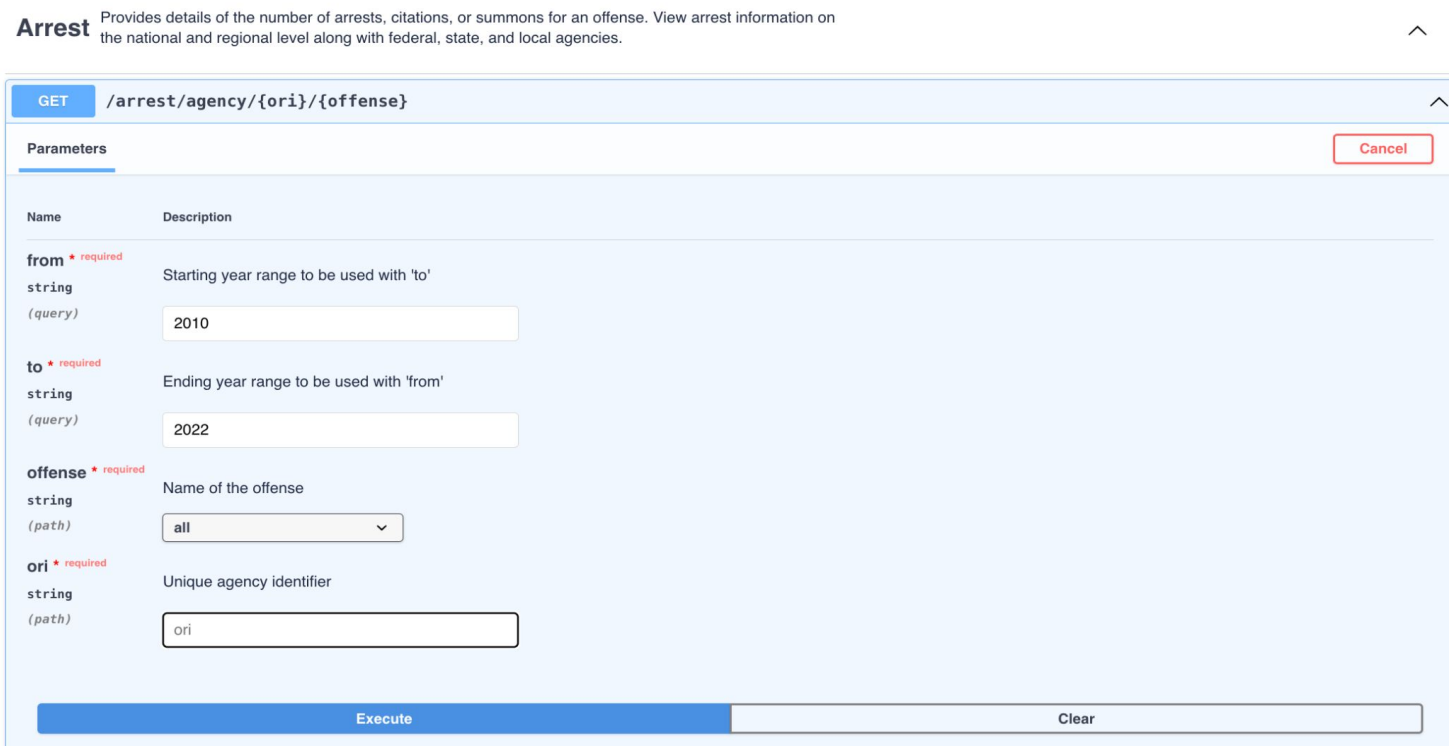


retrieve a list of agencies operating in one state

```
import pandas as pd

agency_data = get_data_from_uri("https://api.usa.gov/crime/fbi/cde/agency/byStateAbbr/GA?API_KEY=KPY
if agency_data:
    agency_df = pd.DataFrame(agency_data)
    print(agency_df.head(3))
```

```
         ori                        agency_name  agency_id state_name  \
0  GA0010000  Appling County Sheriff's Office        3468    Georgia
1  GA0010100         Baxley Police Department        3469    Georgia
2  GA0010300                          Graham       26899    Georgia

  state_abbr   division_name region_name region_desc county_name  \
0         GA  South Atlantic       South  Region III     APPLING
1         GA  South Atlantic       South  Region III     APPLING
2         GA  South Atlantic       South  Region III     APPLING

  agency_type_name  nibrs          nibrs_start_date   latitude   longitude
0           County  False  2019-10-01T00:00:00.000Z  31.782509   -82.35828
1             City   True  2018-12-01T00:00:00.000Z  31.739712  -82.290103
2             City  False                      None       None        None
```

iterate through each agency to retrieve incident data for each year from 2010 to 2022



iterate through each agency to retrieve incident data for each year from 2010 to 2022

Ensure no duplicate agencies and count unique agency entries. Using pandas

662 fbi agencies.

- Loop through each agency to retrieve incident data for each year from 2010 to 2022.
- It gets each agency's yearly data in separate files.

Merge data:

Combine all individual agency files into one dataset.

Concatenate the data from all files and save

```
## Get arrest data for each agency
import time
arrest_df_all = pd.DataFrame()

for ori in agency_df.ori.unique():
    arrest_data = get_data_from_uri("https://api.usa.gov/crime/fbi/cde/arrest/agency/"+ori+"/all?from=2010&
    if arrest_data:
        arrest_data = arrest_data['data']
        arrest_df = pd.DataFrame(arrest_data)
        arrest_df['ori'] = ori
        arrest_df_all = pd.concat([arrest_df_all, arrest_df], axis=0)
        time.sleep(3)
```

Display the data in a readable format

arrest_df_all.to_csv('arrests.csv', index=False)

data_year	Aggravated Assault	All Other Offenses (Except Traffic)	Arson	Burglary	Curfew and Loitering Law Violations	Disorderly Conduct	Driving Under the Influence	Drug Abuse Violations - Grand Total	Drunkenness	Embezzlement	Forgery and Counterfeiting	Fraud	Gambling - Total	Human Trafficking - Commercial Sex Acts	Human Trafficking - Involuntary Servitude	Larceny - Theft	Liquor Laws	Manslaughter by Negligence	Motor Vehicle Theft	Murder and Nonnegligent Manslaughter	Offenses Against the Family and Children	Prostitution and Commercialized Vice	Rape	Robbery	Simple Assault	Stolen Property; Buying, Receiving, Possessing	Suspicion	Vagrancy	Vandalism	Weapons; Carrying, Possessing, Etc.	Sex Offenses (Except Rape, and Prostitution and Commercialized Vice)	ori
2011.0	7.0	110.0	0.0	13.0	0.0	0.0	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2012.0	10.0	66.0	3.0	12.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2013.0	3.0	31.0	0.0	16.0	0.0	0.0	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2014.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2016.0	6.0	52.0	0.0	33.0	0.0	0.0	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2017.0	8.0	89.0	2.0	14.0	0.0	0.0	11.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2018.0	14.0	160.0	0.0	16.0	0.0	0.0	27.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2019.0	6.0	148.0	1.0	5.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2020.0	1.0	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010000
2010.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2011.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2012.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2013.0	1.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2014.0	0.0	0.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2015.0	0.0	2.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2016.0	0.0	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2017.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2018.0	12.0	83.0	0.0	10.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2019.0	10.0	78.0	0.0	7.0	0.0	0.0	11.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2020.0	0.0	15.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2021.0	2.0	59.0	0.0	11.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101
2022.0	8.0	131.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	GA0010101

data_year	Aggravated Assault	All Other Offenses (Except Traffic)	Arson	Burglary	Curfew and Loitering Law Violations	Disorderly Conduct	Driving Under the Influence	Drug Abuse Violations - Grand Total	Drunkenness	Embezzlement	Forgery and Counterfeiting	Fraud	Gambling - Total	Human Trafficking - Commercial Sex Acts	Human Trafficking - Involuntary Servitude	Larceny - Theft	Liquor Laws	Manslaughter by Negligence	Motor Vehicle Theft	Murder and Nonnegligent Manslaughter	Offenses Against the Family and Children	Prostitution and Commercialized Vice	Rape	Robbery	Simple Assault	Stolen Property; Buying, Receiving, Possessing	Suspicion	Vagrancy	Vandalism	Weapons; Carrying, Possessing, Etc.	Sex Offenses (Except Rape, and Prostitution and Commercialized Vice)	ori

Reverse geocode：Converting Coordinates to Zip Codes

-Reverse geocoding is the process of converting geographical coordinates (latitude, longitude) into a readable address or zip code.It's crucial for linking spatial data to specific regions.

-Extract coordinates from the dataset containing incident data

- create new dataframe that doesn't have None value in latitude and longitude by dropping rows if the column latitude and longitude have null empty values

agency_df_latlng = agency_df.dropna(subset=["latitude","longitude"])

Agency_df_latlng.shape

//629

```
# pip install arcgis. To install library
from arcgis.gis import GIS
from arcgis.geocoding import reverse_geocode
from arcgis.geometry import Point
```

Call the function to get zipcode values for all of the agency name

```
def get_zipcode_from_latlng(latitude, longitude):
    gis = GIS()
    point = Point({"X": longitude, "Y":latitude})
    location = reverse_geocode(point)
    zipcode = location['address']['Postal']
    return zipcode
```

Append the zip codes and create a new column in the agency_df_latlng dataframe

```
## Let's call that function to get zipcode value for all of t
zipcodes = []

# We will iterate through the new dataframe where we deleted
for index, row in agency_df_latlng.iterrows():
    latitude = row['latitude']
    longitude = row ['longitude']
    zipcode = get_zipcode_from_latlng(latitude, longitude)
    zipcodes.append(zipcode)

print(len(zipcodes))
```

```
629
```

Burglary in ZipCode 31513

How many total Burglary happened in zipcode 31513

[22]:
```
arrest_31513 = arrest_df.loc[(arrest_df['ori'] == 'GA0010000') | (arrest_df['ori'] == 'GA0010100')]
```

[23]:
```
arrest_31513
```

[23]:

	data_year	Aggravated Assault	All Other Offenses (Except Traffic)	Arson	Burglary	Curfew and Loitering Law Violations	Disorderly Conduct	Driving Under the Influence	Drug Abuse Violations - Grand Total	Drunkenness	...	Rape	Robbery	Simple Assault	Stolen Property: Buying, Receiving, Possessing	Susp
0	2011.0	7.0	110.0	0.0	13.0	0.0	38.0	97.0	51.0	5.0	...	0.0	0.0	14.0	1.0	
1	2012.0	10.0	66.0	3.0	12.0	0.0	17.0	66.0	36.0	3.0	...	0.0	0.0	17.0	3.0	
2	2013.0	3.0	31.0	0.0	16.0	0.0	16.0	19.0	39.0	2.0	...	1.0	0.0	5.0	0.0	
3	2014.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
4	2016.0	6.0	52.0	0.0	33.0	0.0	18.0	21.0	64.0	0.0	...	1.0	0.0	15.0	0.0	
5	2017.0	8.0	89.0	2.0	14.0	0.0	11.0	24.0	64.0	3.0	...	0.0	7.0	14.0	11.0	
6	2018.0	14.0	160.0	0.0	16.0	0.0	32.0	18.0	117.0	3.0	...	0.0	0.0	14.0	4.0	
7	2019.0	6.0	148.0	1.0	5.0	0.0	16.0	27.0	124.0	5.0	...	0.0	0.0	15.0	5.0	
8	2020.0	1.0	3.0	0.0	1.0	0.0	5.0	5.0	16.0	0.0	...	0.0	0.0	2.0	0.0	
9	2010.0	2.0	0.0	0.0	0.0	0.0	4.0	15.0	2.0	0.0	...	0.0	1.0	7.0	0.0	
10	2011.0	0.0	0.0	0.0	1.0	0.0	0.0	11.0	3.0	0.0	...	0.0	0.0	22.0	0.0	
11	2012.0	1.0	0.0	0.0	3.0	0.0	4.0	6.0	0.0	0.0	...	0.0	0.0	19.0	0.0	
12	2013.0	1.0	0.0	0.0	7.0	0.0	0.0	4.0	6.0	3.0	...	0.0	0.0	4.0	0.0	

Let's find out how many burglary happened in zipcode 31513

[25]:
```
arrest_31513.groupby('data_year')['Burglary'].sum()
```

[25]:
```
data_year
2010.0     0.0
2011.0    14.0
2012.0    15.0
2013.0    23.0
2014.0     9.0
2015.0     0.0
2016.0    34.0
2017.0    17.0
2018.0    26.0
2019.0    12.0
2020.0     3.0
2021.0     3.0
2022.0     2.0
Name: Burglary, dtype: float64
```

[26]:
```
arrest_df.shape
```

[26]:
```
(5664, 33)
```

[27]:
```
#arrest_df
#ori_zip_df


arrest_zip_df = pd.merge(arrest_df, ori_zip_df, on='ori')
arrest_zip_df.head()
```

[27]:

Visualizing Assault Data by Zip Code

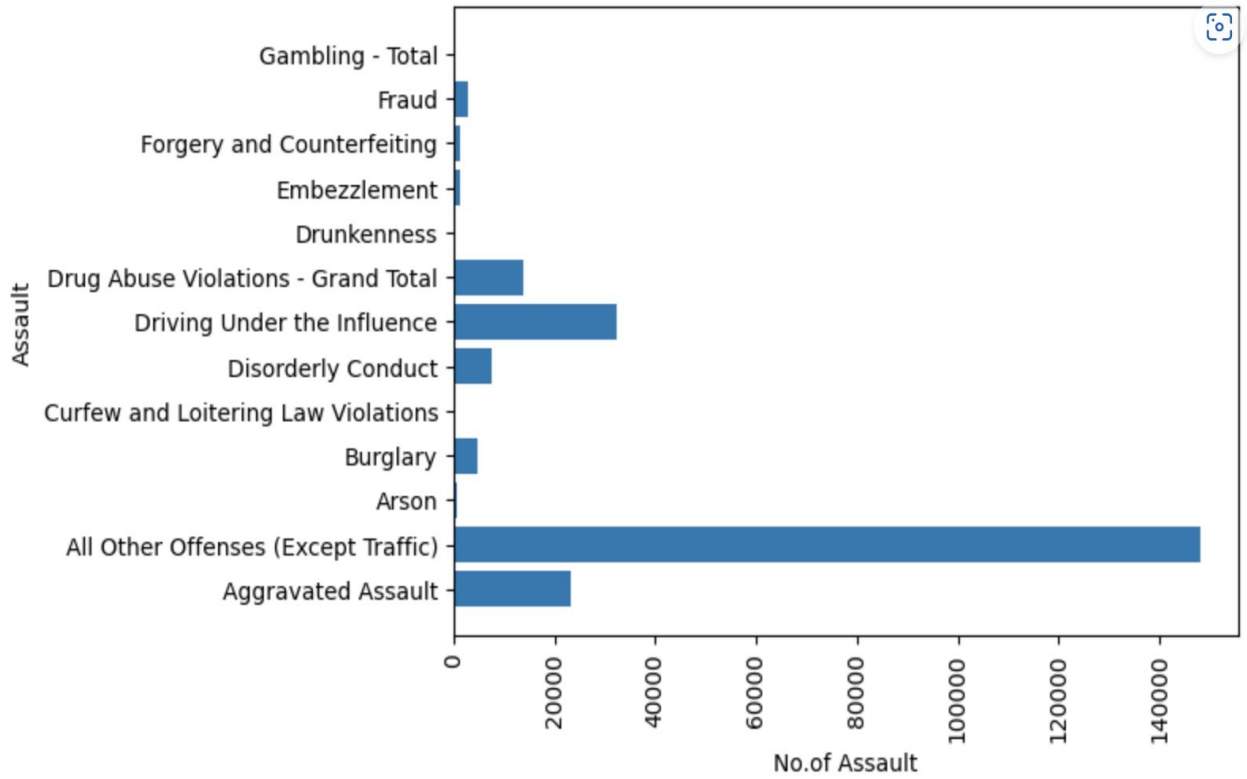
```
import matplotlib.pyplot as plt
```

```
def get_assault_by_zipcode(zipcode):
    data = fbi_df.iloc[:,0:15]
    data_assault_sum = data[data['zip']==zipcode].sum()
    data_assault_sum = data_assault_sum[2:]
    plt.barh(data_assault_sum.index,data_assault_sum.values)
    plt.xlabel("No.of Assault")
    plt.ylabel("Assault")
    plt.xticks(rotation=90)
    plt.show()
```

```
zip_input = input("Input your zipcode")
if int(zip_input) in fbi_df['zip'].values:
    get_assault_by_zipcode(int(zip_input))
else:
    print("We don't have information about that zipcode")
```

Assaults happened in zip 99577

```
Input your zipcode 99577
```



Relationship between homeprices and crime data and education

- ML
  - Correlation matrix
  - Model - training : randomforestregressor (mse)
  - Normalization: Min_max_scaler

Education Dataset

Launcher | 2022_education.csv

Delimiter: ,

	Geography	zip	Less than high school…	High school graduate …	Some college or asso…	Bachelor degree or hi…	Graduate degr
1	860Z200US00601	601	4044	21001	3364	6025	
2	860Z200US00602	602	9648	45695	7510	17381	
3	860Z200US00603	603	9360	68058	8954	24178	
4	860Z200US00606	606	1710	6420	991	1163	
5	860Z200US00610	610	5378	33781	5143	12085	
6	860Z200US00611	611	345	1659	213	651	
7	860Z200US00612	612	9034	90601	13731	36863	
8	860Z200US00616	616	1707	14562	1974	4619	
9	860Z200US00617	617	3829	31742	4541	10652	
10	860Z200US00622	622	1778	10282	1890	3743	
11	860Z200US00623	623	7097	52386	9050	23891	
12	860Z200US00624	624	4260	26835	4309	10818	
13	860Z200US00627	627	6928	41272	8259	14450	
14	860Z200US00631	631	392	1597	267	937	
15	860Z200US00636	636	157	1074	214	408	
16	860Z200US00637	637	3593	31602	4489	11765	
17	860Z200US00638	638	4461	21821	3651	5771	
18	860Z200US00641	641	4902	31595	6159	10927	
19	860Z200US00646	646	4786	50247	7728	23231	
20	860Z200US00647	647	748	6433	765	1477	
21	860Z200US00650	650	2370	19014	3119	5557	
22	860Z200US00652	652	634	5489	962	2384	

Crime Dataset

Launcher | fbi_zip_crime_aggregate_w | +

Delimiter: ,

	zip	Year	Aggravated Assault	All Other Offenses (E…	Arson	Burglary	Curfew and Lo
1	99577	2010	1902.0	17672.0	44.0	312.0	
2	99577	2011	1690.0	16744.0	58.0	290.0	
3	99577	2012	1906.0	15752.0	44.0	334.0	
4	99577	2013	1578.0	13892.0	28.0	252.0	
5	99577	2014	1640.0	13284.0	38.0	294.0	
6	99577	2015	1732.0	13570.0	66.0	330.0	
7	99577	2016	1838.0	11402.0	44.0	384.0	
8	99577	2017	2038.0	11458.0	26.0	490.0	
9	99577	2018	2250.0	11000.0	52.0	604.0	
10	99577	2019	2300.0	10624.0	40.0	610.0	
11	99577	2020	2194.0	8904.0	46.0	436.0	
12	99577	2021	0.0	46.0	0.0	0.0	
13	99577	2022	2156.0	3830.0	56.0	488.0	
14	99701	2010	142.0	1402.0	0.0	28.0	
15	99701	2011	28.0	886.0	4.0	24.0	
16	99701	2012	76.0	594.0	0.0	46.0	
17	99701	2013	100.0	656.0	0.0	34.0	
18	99701	2014	100.0	804.0	0.0	36.0	
19	99701	2015	80.0	574.0	0.0	30.0	
20	99701	2016	88.0	426.0	2.0	66.0	
21	99701	2017	104.0	510.0	2.0	54.0	
22	99701	2018	126.0	648.0	4.0	20.0	

Home Prices Dataset

homeprice_crime_2022.csv

Delimiter: ,

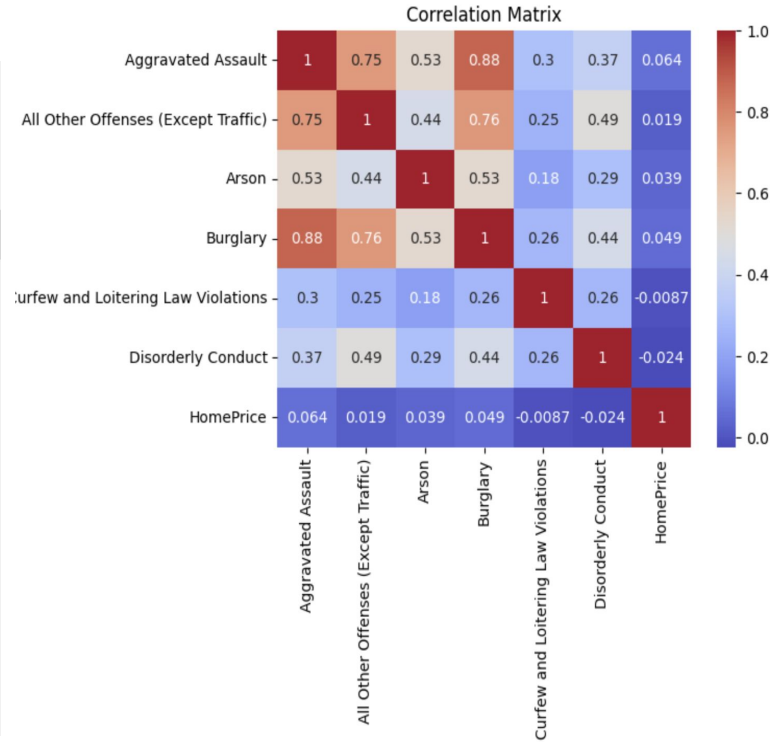
	Vandalism	Weapons: Carrying, P…	Sex Offenses (Except…	State	City	Metro	CountyName	HomePrice
1	1208.0	290.0	296.0	AK	Eagle River	Anchorage, AK	Anchorage Borough	421193.9974632655
2	98.0	54.0	2.0	AK	Fairbanks	Fairbanks, AK	Fairbanks North Star …	236133.5624647099
3	80.0	38.0	12.0	AK	Juneau	Juneau, AK	Juneau Borough	472938.4739336625
4	4.0	6.0	0.0	AK	Ketchikan	Ketchikan, AK	Ketchikan Gateway B…	386640.9335621988
5	12.0	10.0	0.0	AK	Kodiak		Kodiak Island Borough	459772.0736162829
6	6.0	0.0	0.0	AK	Sitka		Sitka Borough	588573.533119711
7	0.0	0.0	0.0	AK	Wrangell		Wrangell Borough	298297.94434762566
8	0.0	0.0	0.0	AK	Palmer	Anchorage, AK	Matanuska Susitna B…	366367.9261497548
9	2.0	4.0	0.0	AK	Soldotna		Kenai Peninsula Boro…	339197.36588897585
10	0.0	0.0	0.0	AK			Haines Borough	290877.96476889827
11	10.0	4.0	2.0	AK	Homer		Kenai Peninsula Boro…	366673.0225532467
12	32.0	12.0	0.0	AK	Kenai		Kenai Peninsula Boro…	310672.5213491242
13	0.0	0.0	0.0	AK	Fairbanks	Fairbanks, AK	Fairbanks North Star …	298939.9255060815
14	4.0	6.0	0.0	AK	North Pole	Fairbanks, AK	Fairbanks North Star …	308285.47326387296
15	0.0	0.0	0.0	AK	Salcha	Fairbanks, AK	Fairbanks North Star …	200701.1376864817
16	346.0	744.0	6.0	AL	Forestdale	Birmingham-Hoover, AL	Jefferson County	141848.62442317654
17	36.0	66.0	0.0	AL	Bessemer	Birmingham-Hoover, AL	Jefferson County	71234.39997309867
18	4.0	12.0	0.0	AL	Mountain Brook	Birmingham-Hoover, AL	Jefferson County	705799.8900266549
19	4.0	0.0	0.0	AL	Tarrant	Birmingham-Hoover, AL	Jefferson County	84417.72930661887
20	0.0	4.0	0.0	AL	Homewood	Birmingham-Hoover, AL	Jefferson County	461115.1830919778
21	12.0	64.0	2.0	AL	Vestavia Hills	Birmingham-Hoover, AL	Jefferson County	369581.0854756904
22	6.0	6.0	0.0	AL	Irondale	Birmingham-Hoover, AL	Jefferson County	252691.43169859247
23	2.0	34.0	0.0	AL	Pleasant Grove	Birmingham-Hoover, AL	Jefferson County	220502.95919913705
24	8.0	2.0	0.0	AL	Birmingham	Birmingham-Hoover, AL	Jefferson County	217615.0403134825
25	0.0	0.0	0.0	AL	Center Point	Birmingham-Hoover, AL	Jefferson County	155345.0079095897
26	2.0	2.0	0.0	AL	Trussville	Birmingham-Hoover, AL	Jefferson County	359571.22748729965
27	70.0	72.0	2.0	AL	Mobile	Mobile, AL	Mobile County	214856.31647636663

Correlation Matrix

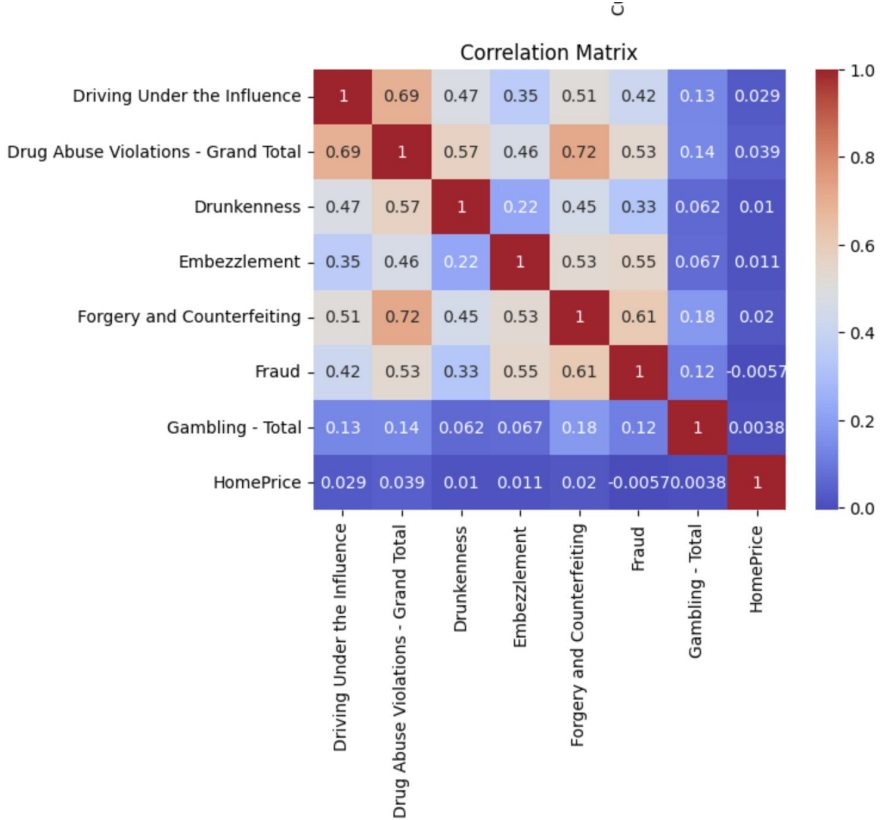
Let's find out the relationship between different assault and the home prices

[26]:

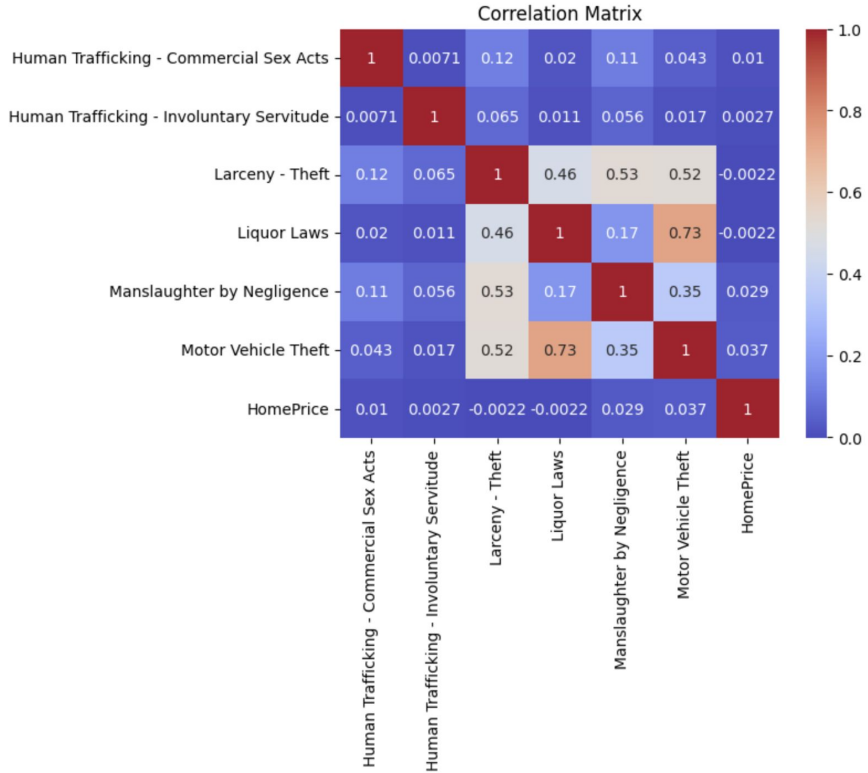
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Correlation matrix
# The matrix is a table in which every cell contains a correlation coefficient, where 1 is considered a strong relationship between variables, 0 a
correlation_matrix = fbi_df[['Aggravated Assault',
       'All Other Offenses (Except Traffic)', 'Arson', 'Burglary',
       'Curfew and Loitering Law Violations', 'Disorderly Conduct',
       'HomePrice']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()



correlation_matrix = fbi_df[['Driving Under the Influence', 'Drug Abuse Violations - Grand Total',
       'Drunkenness', 'Embezzlement', 'Forgery and Counterfeiting', 'Fraud',
       'Gambling - Total','HomePrice']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()



correlation_matrix = fbi_df[['Human Trafficking - Commercial Sex Acts',
       'Human Trafficking - Involuntary Servitude', 'Larceny - Theft',
       'Liquor Laws', 'Manslaughter by Negligence', 'Motor Vehicle Theft',
       'HomePrice']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



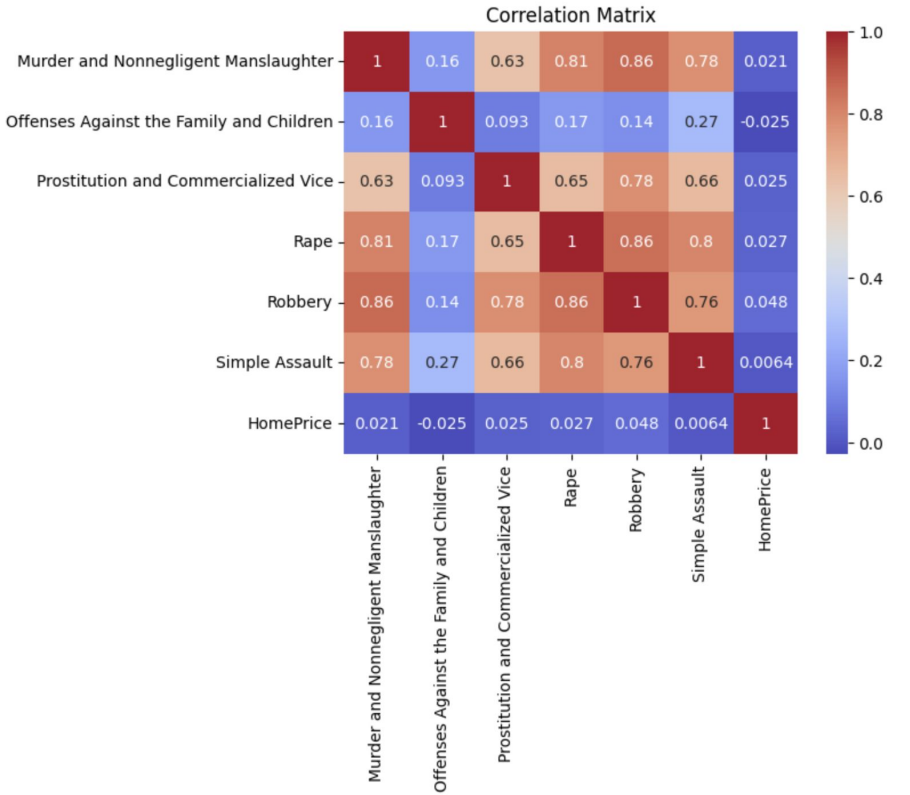
Correlation Matrix



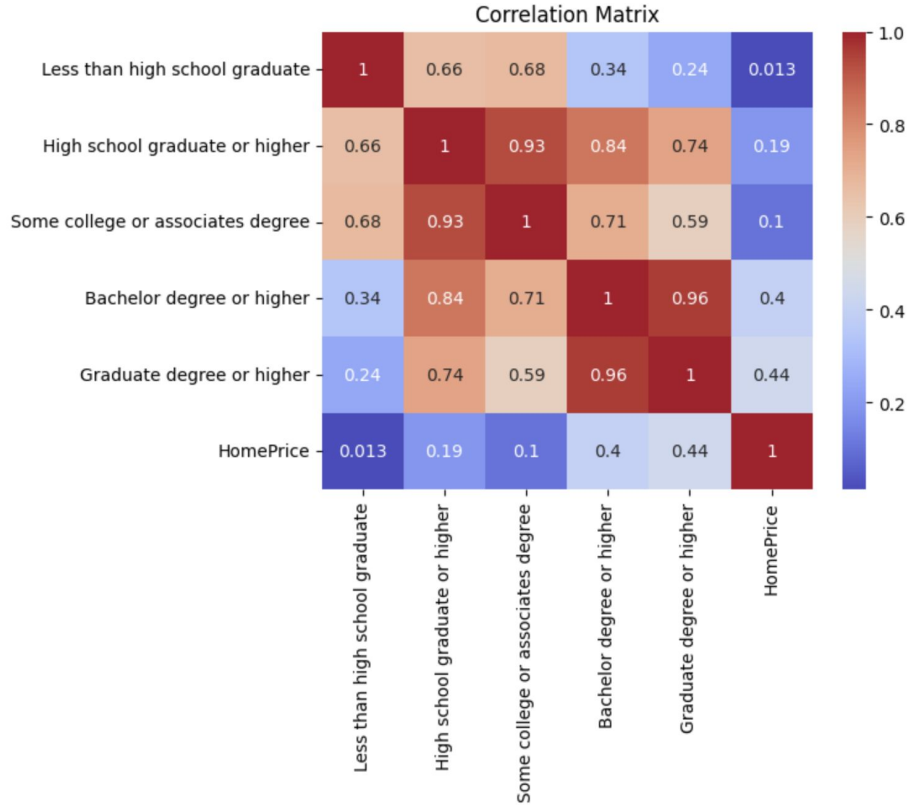
Correlation Matrix



Correlation Matrix



Correlation Matrix



Normalization

```
# Min-Max Normalization
min_max_scaler = MinMaxScaler()
X = pd.DataFrame(min_max_scaler.fit_transform(X), columns=X.columns)
Y = pd.DataFrame(min_max_scaler.fit_transform(Y), columns=Y.columns)
```

RandomForestAggressor

Where:

- β0 is the intercept.
- $\beta 1, \beta 2, \beta 3$ are the coefficients for the independent variables.
- ϵ is the error term.

[27]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import MinMaxScaler, StandardScaler, MaxAbsScaler, RobustScaler

from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import PolynomialFeatures

# Features and target variable
data_X = fbi_df.loc[:,['Aggravated Assault',
       'All Other Offenses (Except Traffic)', 'Arson', 'Burglary',
       'Curfew and Loitering Law Violations', 'Disorderly Conduct',
       'Driving Under the Influence', 'Drug Abuse Violations - Grand Total',
       'Drunkenness', 'Embezzlement', 'Forgery and Counterfeiting', 'Fraud',
       'Gambling - Total', 'Human Trafficking - Commercial Sex Acts',
       'Human Trafficking - Involuntary Servitude', 'Larceny - Theft',
       'Liquor Laws', 'Manslaughter by Negligence', 'Motor Vehicle Theft',
       'Murder and Nonnegligent Manslaughter',
       'Offenses Against the Family and Children',
       'Prostitution and Commercialized Vice', 'Rape', 'Robbery',
       'Simple Assault', 'Stolen Property: Buying, Receiving, Possessing',
       'Suspicion', 'Vagrancy', 'Vandalism',
       'Weapons: Carrying, Possessing, Etc.',
       'Sex Offenses (Except Rape, and Prostitution and Commercialized Vice)']]

data_Y = pd.DataFrame(fbi_df.loc[:,'HomePrice'])
```

Output

```
/userapp/workshop24/venv/lib/python3.9/site-packages/sklearn/base.py:1473: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
Mean Squared Error: 0.0005446086392279777
R-squared: 0.21135295100813822
Predicted House Prices: [0.06342942 0.01762943 0.04974515 ... 0.07129463 0.01664068 0.02324042]
[546323.56862488 162421.62632192 431620.13025358 ... 612250.84908411
 154133.85670767 209453.75318182]
[575952.78051513 118434.93581778 249122.3458304  ... 349674.9363878
  97178.79599571 174437.87010055]
```

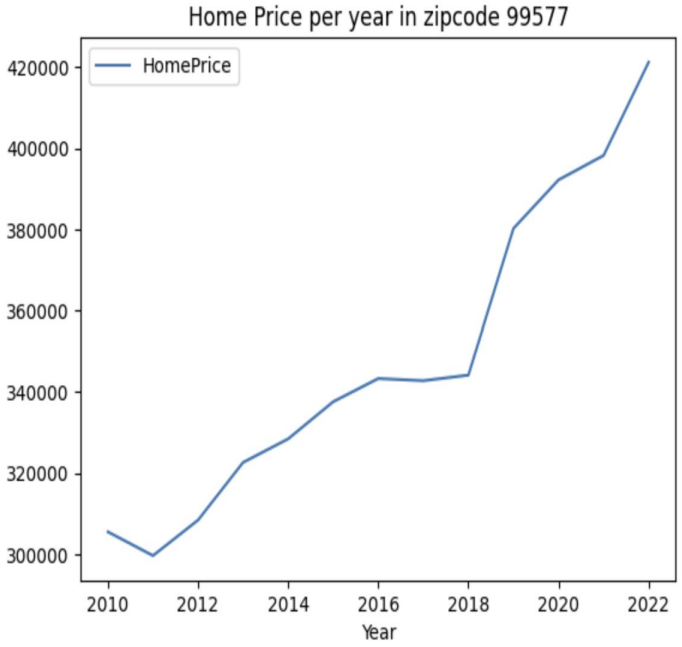
Home

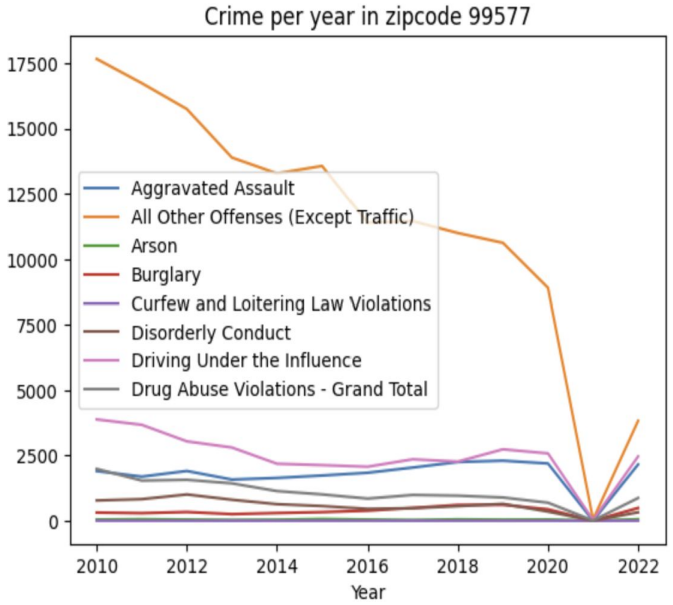
[46]:
```
def plot_homeprice_for_zipcode(zipcode,df):
    year_homeprice_df = df[df['zip']==zipcode]
    year_homeprice_df = year_homeprice_df.loc[:,['HomePrice']]
    year_homeprice_df.plot()
    plt.title("Home Price per year in zipcode "+str(zipcode))
    plt.show()

def plot_crime_for_zipcode(zipcode,df):
    year_homeprice_df = df[df['zip']==zipcode]
    year_homeprice_df = year_homeprice_df.loc[:,['Aggravated Assault','All Other Offenses (Except Traffic)','Arson','Burglary',
                                'Curfew and Loitering Law Violations','Disorderly Conduct','Driving Under the Influence',
                                'Drug Abuse Violations - Grand Total']]
    year_homeprice_df.plot()
    plt.title("Crime per year in zipcode "+str(zipcode))
    plt.show()
```

[47]:
```
zip_input = input("Input your zipcode")
if int(zip_input) in year_homeprice_df['zip'].values:
    plot_homeprice_for_zipcode(int(zip_input), year_homeprice_df)
    plot_crime_for_zipcode(int(zip_input), year_homeprice_df)
else:
    print("We don't have information about that zipcode")
```

```
Input your zipcode 99577
```

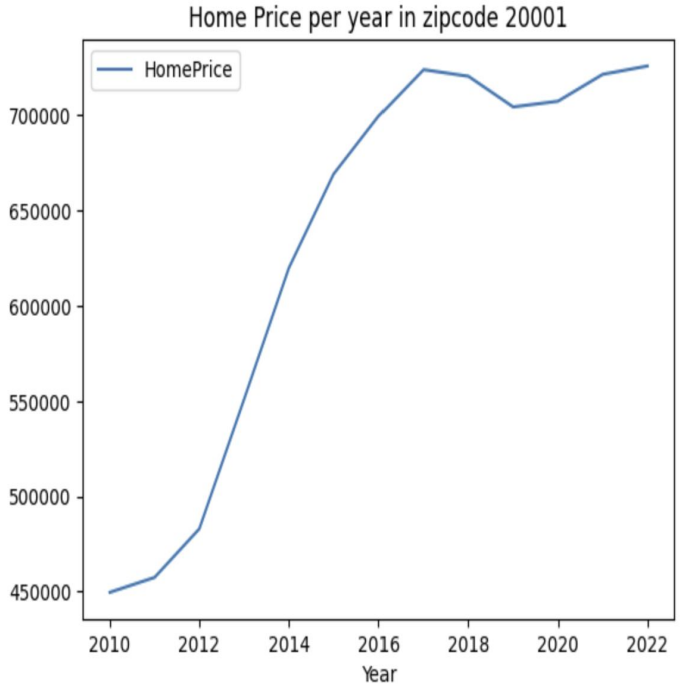
Home Price Rate and Crime Rate from 2010 to 2022 in zipcode 99577

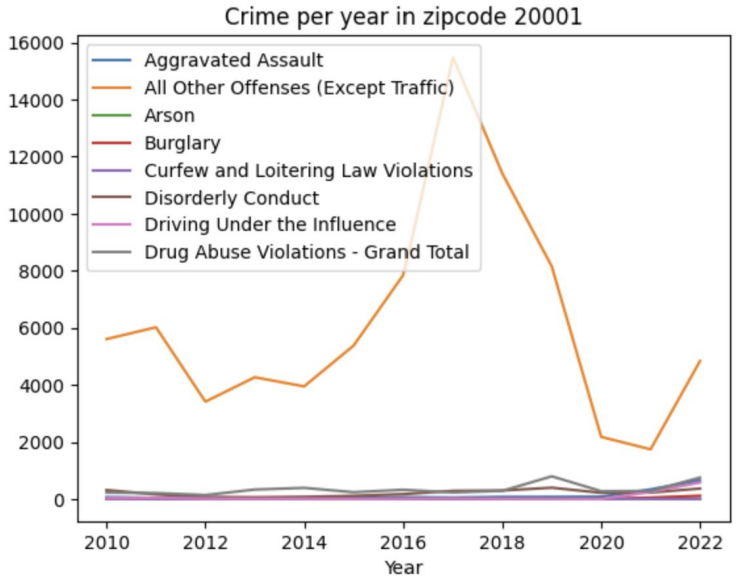




Home Price Rate and Crime Rate from 2010 to 2022 in zipcode 20001

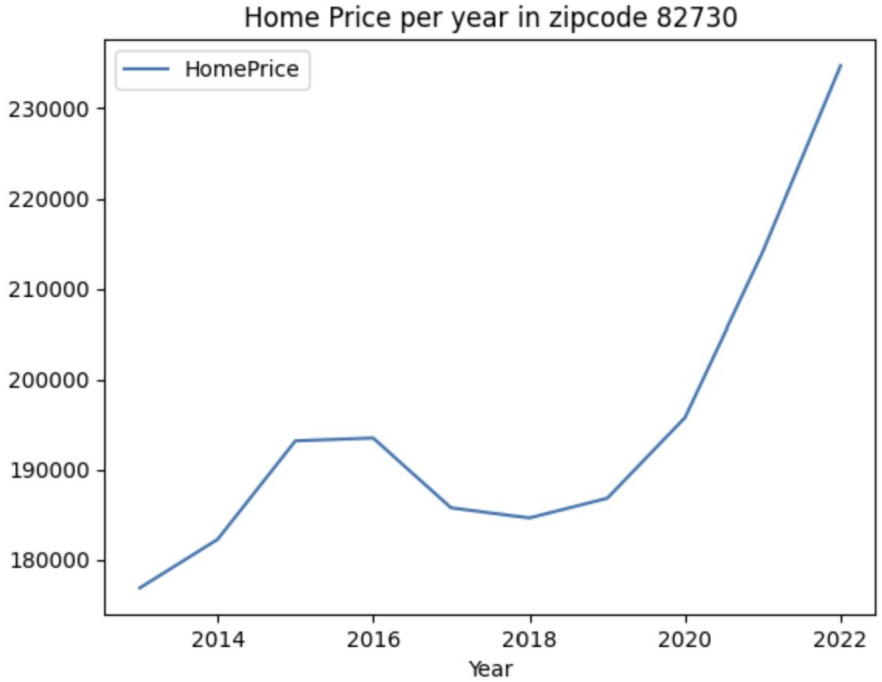
```
Input your zipcode 20001
```

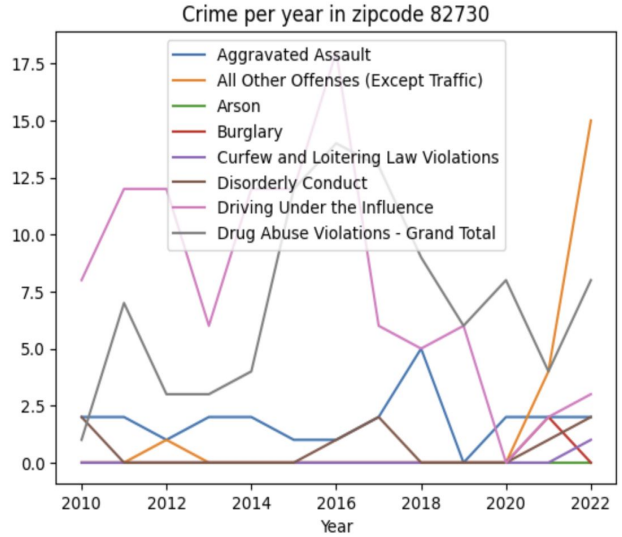




Home Price Rate and Crime Rate from 2010 to 2022 in zipcode 82730

```
Input your zipcode 82730
```





What we learned

API

Python

Numpy

Pandas

Sklearn

Matplotlib

GIS

Challenges

- Get all the agency arrest data into one file
- How to clean data
- Debug the code
- Find the best model to predict data. Linear regression did not work.

Thank You!