Insights for Chicago to Improve Public Safety

Implementation Report

Bekalu Tadesse | CSCI622

1. Introduction

This project aims to support data-driven public safety initiatives for Chicago by applying data engineering principles, leveraging public datasets from data.gov, cloud and technology.

Through data ingestion, transformation, and analysis, the project aims to uncover valuable insights that can contribute to improving public safety.

The analysis seeks to provide actionable insights to policymakers, enabling them to identify specific geographic areas that require attention and make informed decisions to enhance public safety effectively.

2. Project Setup and Architecture

2.1 Overall architecture

The project ingests 4 datasets (2 dynamic and 2 static) from data.gov API's into azure cloud blob storage (object storage). Ingestion is carried out using a python code that runs in a pip env virtual environment on a local machine.

Exploratory data analysis and Transformation is done within Azure databricks to unify datasets, merge and aggregate data. The ingestion code as well as Azure databricks transformation code is synched to a GitHub repository.

The final step of analysis/serving is done using PowerBI and Jupyter Notebook.

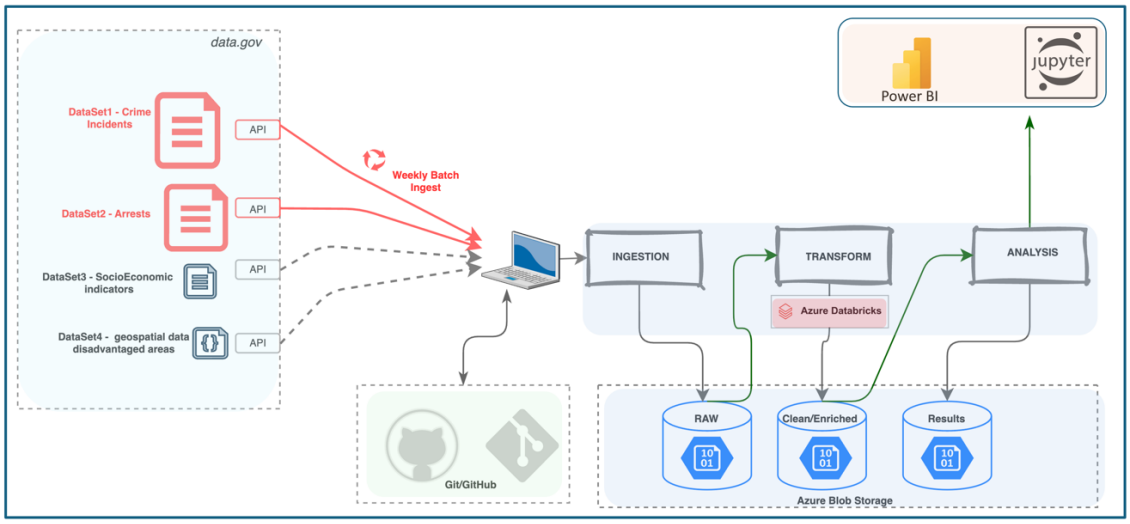


Figure 1 - Overall architecture

Azure Cloud Setup

Within azure cloud a storage account, key vault and Azure data bricks workspace is created and configured using point and click. Automation of this task will be done in the future to improve this project further.

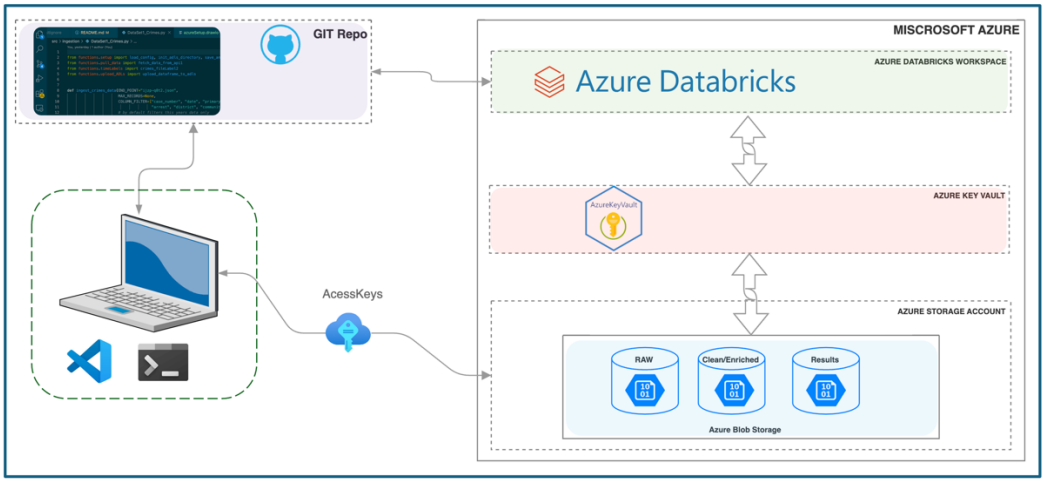


Figure 2 - Azure Cloud Setup

Three Blob Containers—designated for raw data, transformed data, and results—are **programmatically** created within the Azure storage account using Python code responsible for data ingestion. During the ingestion process, data is first retrieved from the API, saved as CSV files on a local machine as an intermediate step, and then uploaded to the respective containers in the Azure storage account.

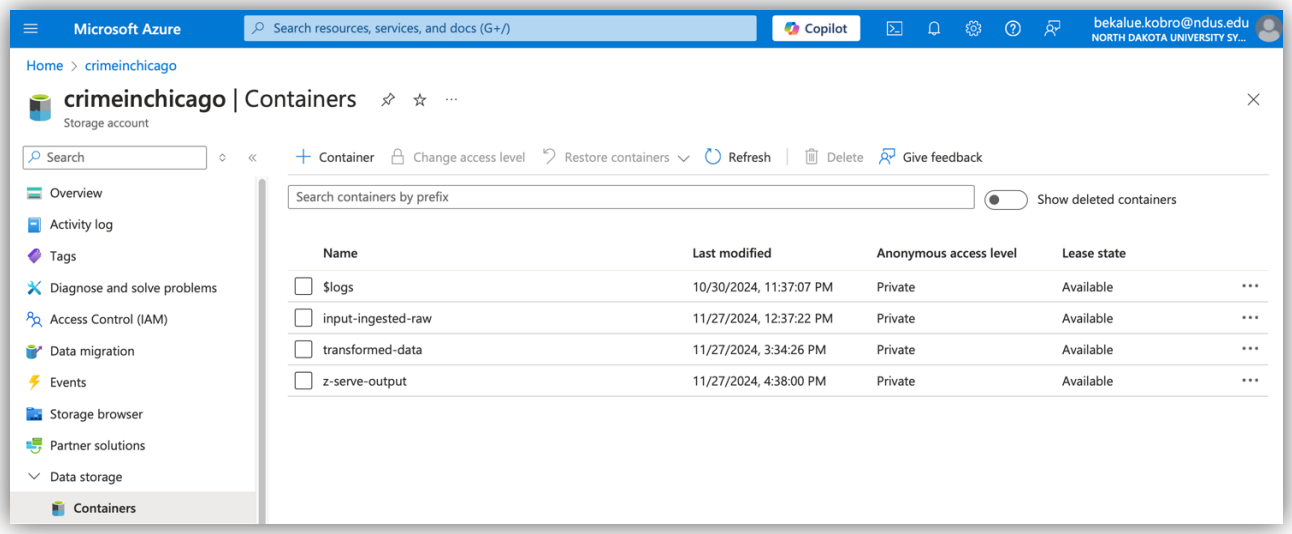


Figure 3 - blob containers

Local machine setup (virtual environment)

A **Pipenv** virtual environment has been created on the local machine to run the ingestion code. The necessary Python packages are installed within this environment to ensure all dependencies are met for the process.

The **Pipfile** or **Pipfile.lock** can be used to recreate the development environment. Some of python packages installed and their use case is shown on the table below.

Python package	Use Case
azure-storage-file-datalake	- Programmatic ingestion to azure blob storage
tqdm	- Show progress bar during ingestion
Folium, geopandas	- visualize geospatial data
Plotly, matplotlib	- visualizations within jupyter notebook

3. Implementation Steps (Road map)

The implementation follows the data engineering lifecycle, consisting of three main steps: ingestion, transformation, and analysis/serving.

During the **ingestion** phase, data is retrieved from the Data.gov API and stored in Azure Blob Storage. In the **transformation** phase, the data is unified, merged, and aggregated to prepare it for the next step. In the **analysis** phase, the transformed data is used for analysis in Power BI and Jupyter Notebook.

In **Jupyter Notebook**, the aggregated and merged data is used to create visualizations, while in **Power BI**, the merged data is utilized to build dashboards.

3.1 Ingestion

3.1.1 Data sources:

Four data sources has been used in this project. These are:

DataSource1: Crimes	- Dynamic data source - updated frequently - shows crime incidents - 8.19 rows x 22 columns
DataSource2: Arrests	- Dynamic data source

	- updated frequently - shows arrests - 660k rows x 24 rows
DataSource3: Socio-economic indicators	- Static data source - updated less frequently - shows economic indicators such as education, employment, poverty, etc. - 78 rows x 9 columns
DataSource4: Socioeconomically disadvantaged areas	- Static data source (geospatial data) - updated less frequently - shows areas in Chicago that have poor socio economic indicators - s254 rows x 1 column

3.2 Ingestion initial steps:

Create API Key & Azure Acess:

- Register on Data.gov and create an API key using the "SignUpforAppToken" option.
- Store the API key in a config file.
- Also copy SAS key into another config file to be able to write to azure storage programmatically.

Data Retrieval via API

- Pull data via the Data.gov API.
- **Libraries Required:** Install pandas, soapy, jupyter, and azure-storage-file-datalake.

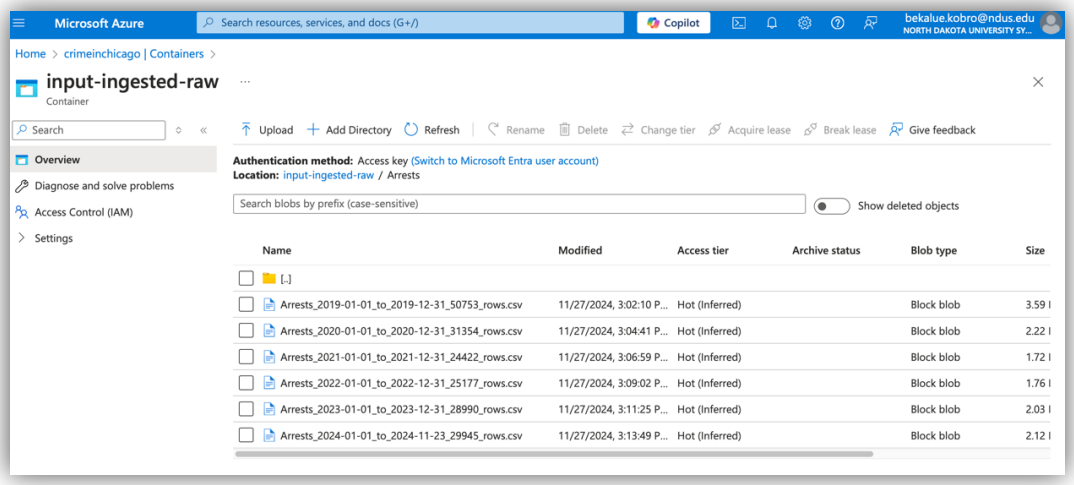
Programmatic Storage

- Create a directory within a container in Azure Blob Storage programmatically.
- Store datasets in .csv format for easy access in the next step.

3.2.1 Dynamic data Ingestion: Batch and row/column Filter

A batch ingestion technique is employed to handle dynamic data sources. Since these sources include a significant number of records (over 8 million, and 600k in data sets 1 and 2 respectively) a row and column filters are applied at the API level. This approach prevents API throttling and reduces the need for extensive data cleaning later.

Column filters are used to retrieve only the relevant columns, while additional row filters are incorporated into the ingestion code to exclude null records and process the data in smaller, manageable chunks, organized by year as shown on the figures below.



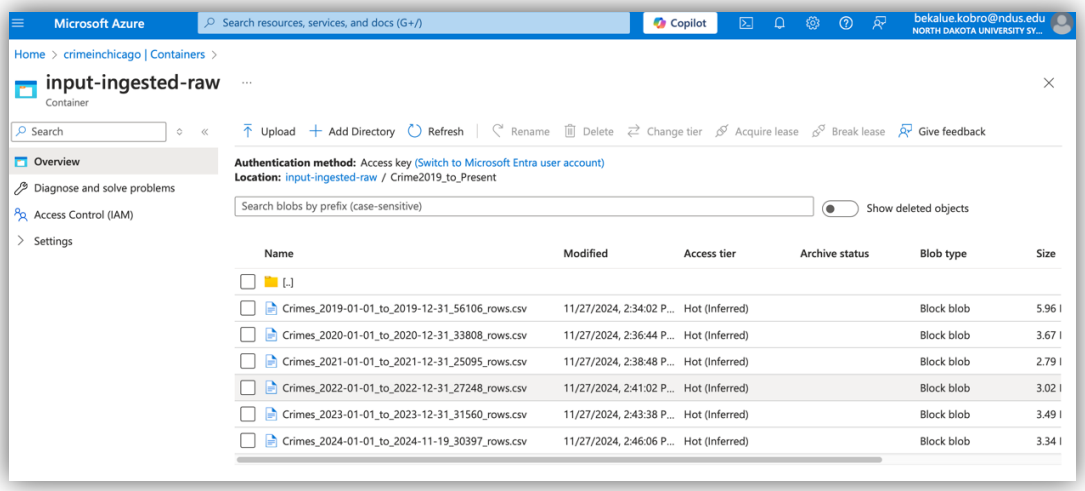


Figure 4 - chunks of data by year after batch ingestion by year

After the historical data is fully retrieved, continuous ingestion is performed by applying row filters with a specified time window. This allows for the dynamic datasets to be ingested weekly as new data becomes available.

3.3 Transformation: Unify, Join, Aggregate data

Data transformation involves unifying the segmented data from each source into unified datasets. These unified datasets are then merged into a single large table using common attributes/columns as the basis for the joins.

Summary of joining attributes for the data sets is listed on the table below:

DataSets	Join_attribute
DataSet1_Crimes & DataSet2_Arrests	case_number
Dataset_2_Arrests & DataSet3_incicators	Community_area_number
Dataset1_crimes **&** **Dataset4_disadvantagedAreas**	Latitude, longitude

Dataset Aggregation:

The merged data is further aggregated based on specific indicator columns to enable deeper analysis within Jupyter Notebook.

Sime Data Transformations with PowerBI:

Simpler data transformations are also carried out within power BI such as adding a new column, and some aggregations using DAX.

Data Lineage:

After ingestion of Historical dynamic data in yearly batches, these yearly segments of data are then consolidated during the transformation phase to produce a unified CSV file. Finally, datasets are joined by common attributes as shown on the high-level data lineage graph below.

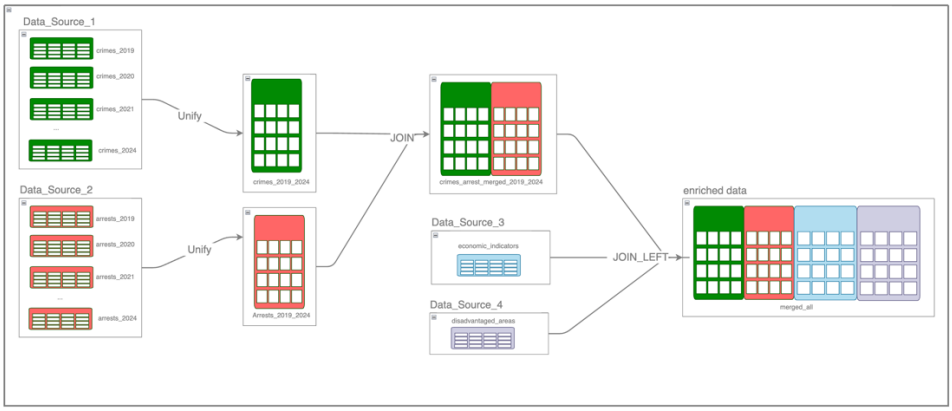


Figure 5 - Data lineage shows changes made to the data after ingestion down to transformation steps

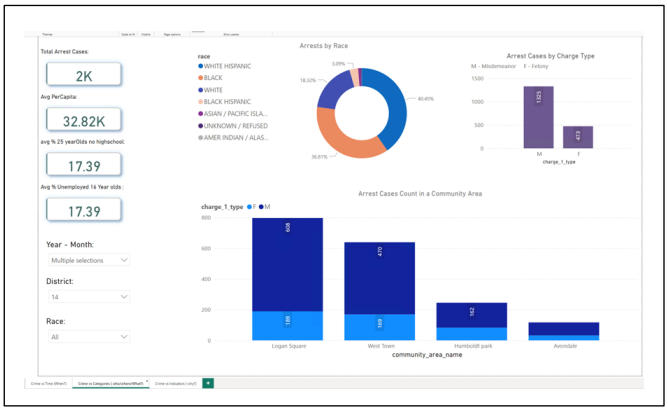
4. Serving/Analysis

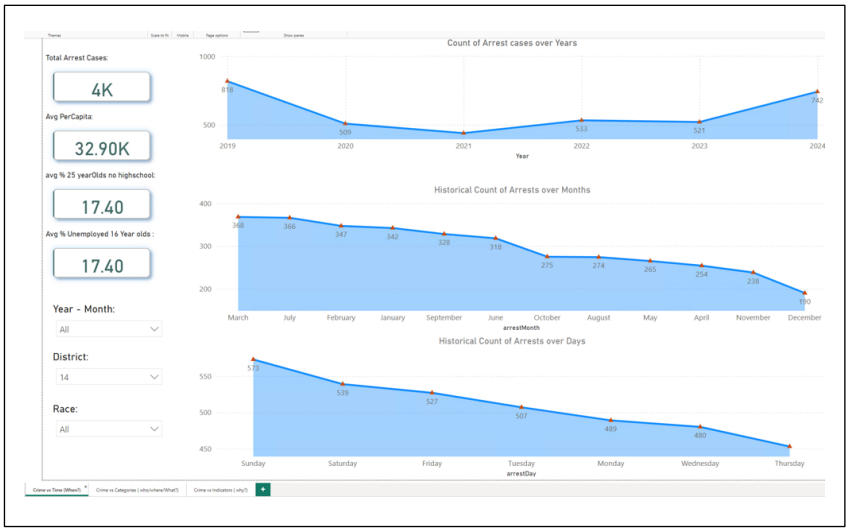
The joined data is stored back into Azure Blob Storage, specifically within the cleaned blob container. From there, the data is imported into Power BI for analysis. Additionally, a copy of the aggregated datasets is loaded into Jupyter Notebook for further analysis, as outlined in the overall technical architecture.

4.1 Guiding Questions for Analysis:

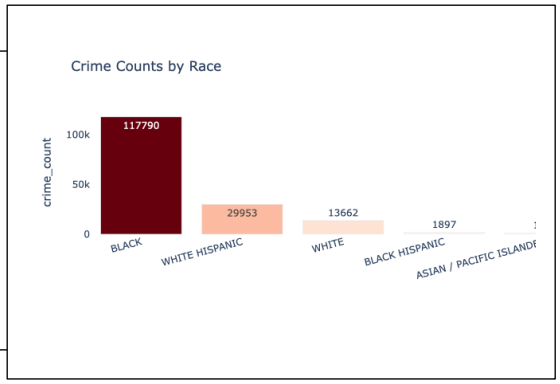
- What areas with high crime rates relative to socioeconomic factors?
- How crime trended over years, Months?
- Is crime more common towards weekends or workdays?
- What are High-risk districts that need more attention?
- Does poor socio economic indicators imply high crime rate?

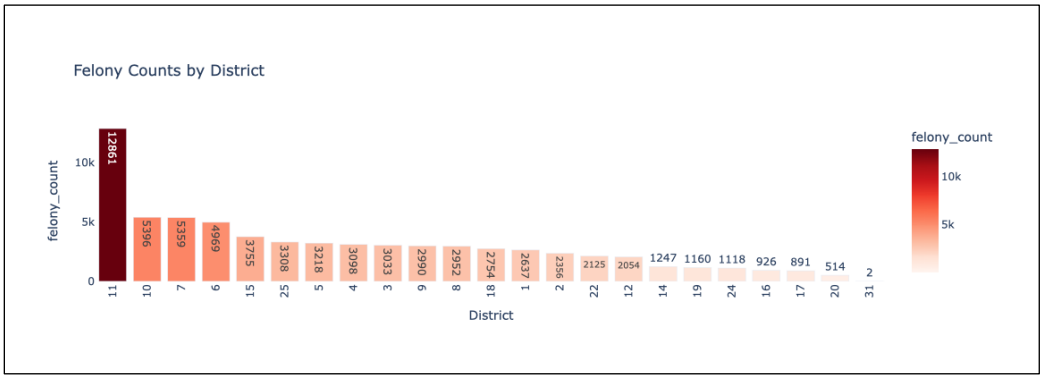
4.2 Visuals from Power BI:

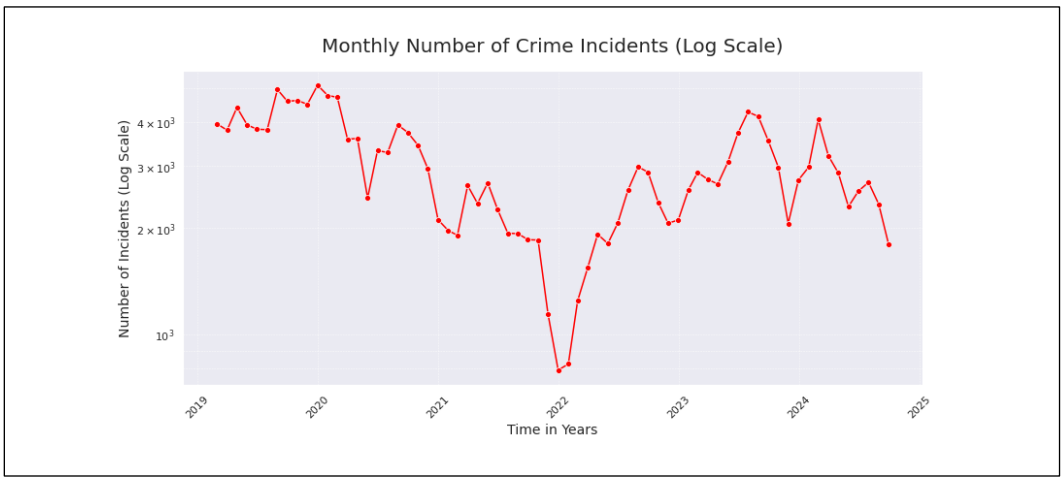


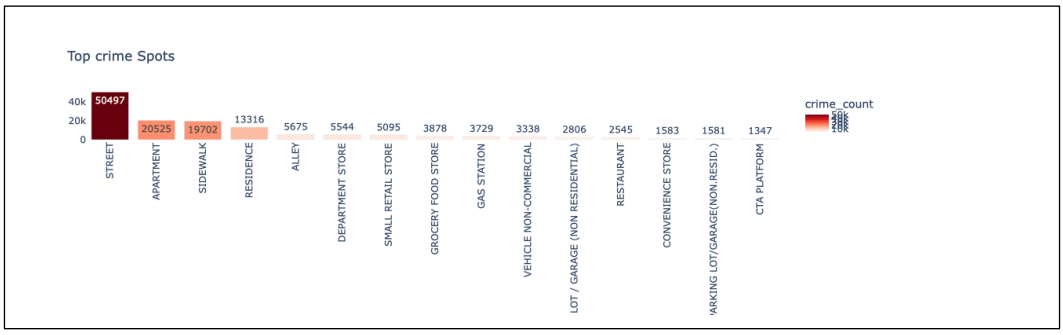


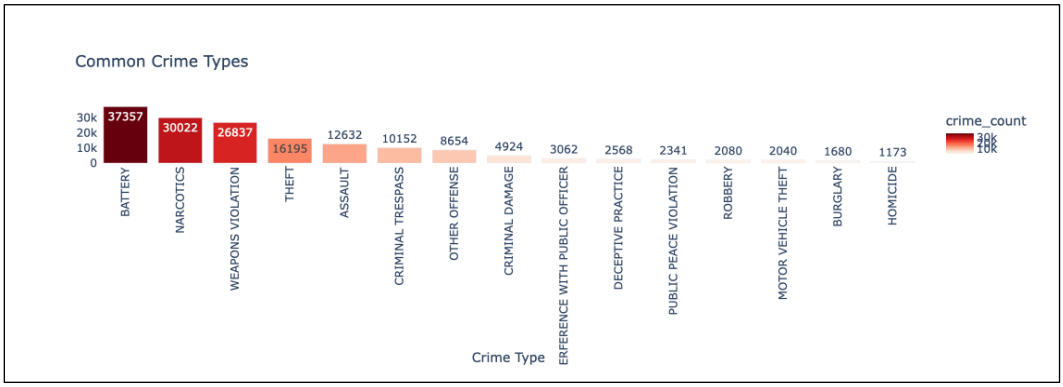
4.3 Visuals from Jupyter Notebook

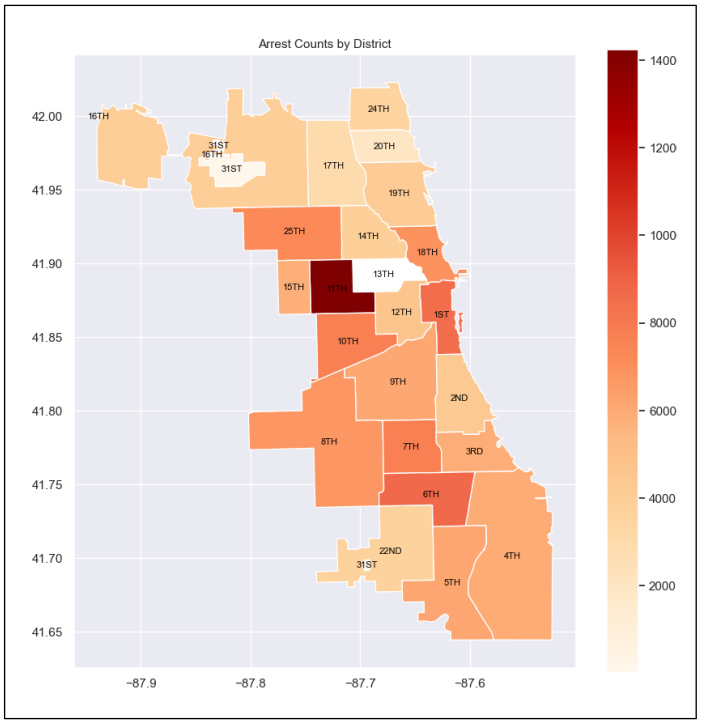


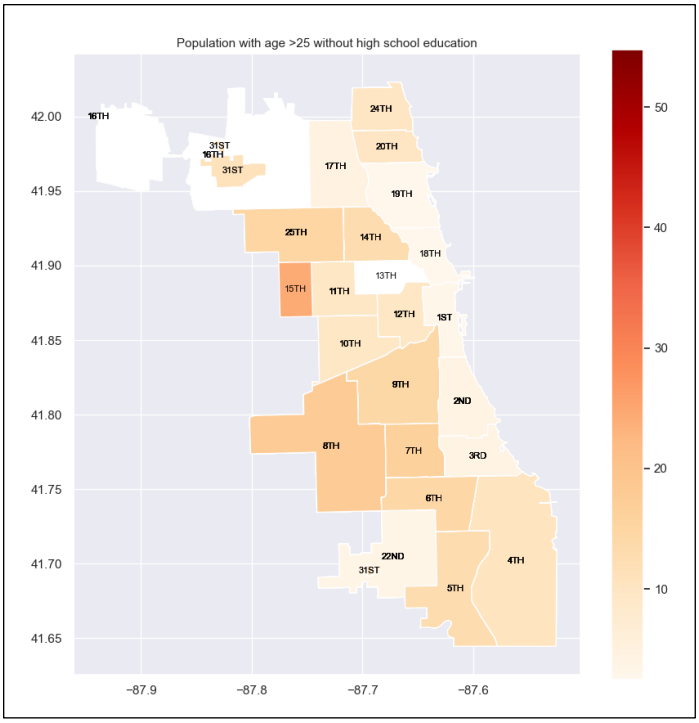

Felony Counts by District
felony_count
10k
5k
0
12861
5396
5359
4969
3755
3308
3218
3086
3033
2990
2952
2754
2637
2356
2125
2054
1247
1160
1118
926
891
514
2
11
10
7
6
15
25
5
4
3
9
8
18
1
2
22
12
14
19
24
16
17
20
31
District
felony_count
10k
5k

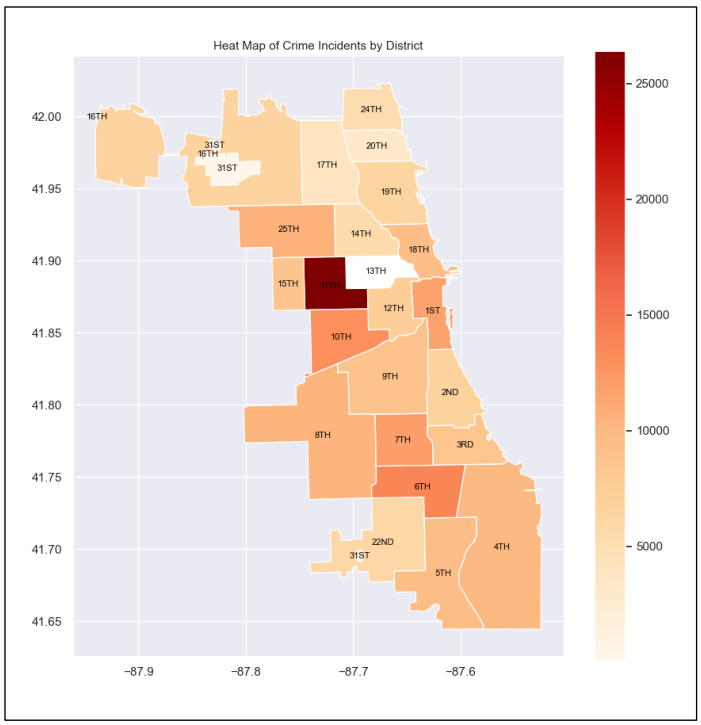

Monthly Number of Crime Incidents (Log Scale)
Number of Incidents (Log Scale)
4 × 10^3
3 × 10^3
2 × 10^3
10^3
2019
2020
2021
2022
2023
2024
2025
Time in Years

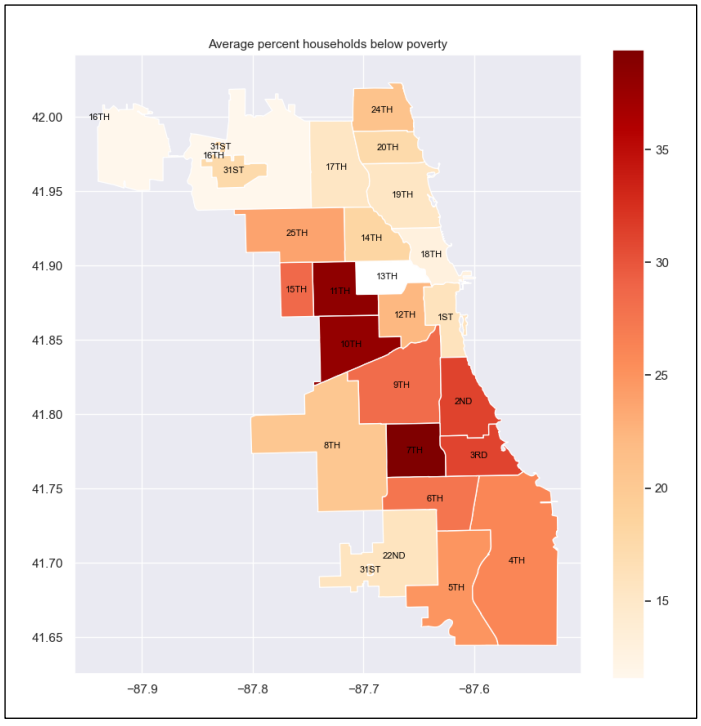

Top crime Spots
40k
20k
0
50497
20525
19702
13316
5675
5544
5095
3878
3729
3338
2806
2545
1583
1581
1347
STREET
APARTMENT
SIDEWALK
RESIDENCE
ALLEY
DEPARTMENT STORE
SMALL RETAIL STORE
GROCERY FOOD STORE
GAS STATION
VEHICLE NON-COMMERCIAL
LOT / GARAGE (NON RESIDENTIAL)
RESTAURANT
CONVENIENCE STORE
'ARKING LOT/GARAGE(NON.RESID.)
CTA PLATFORM
crime_count

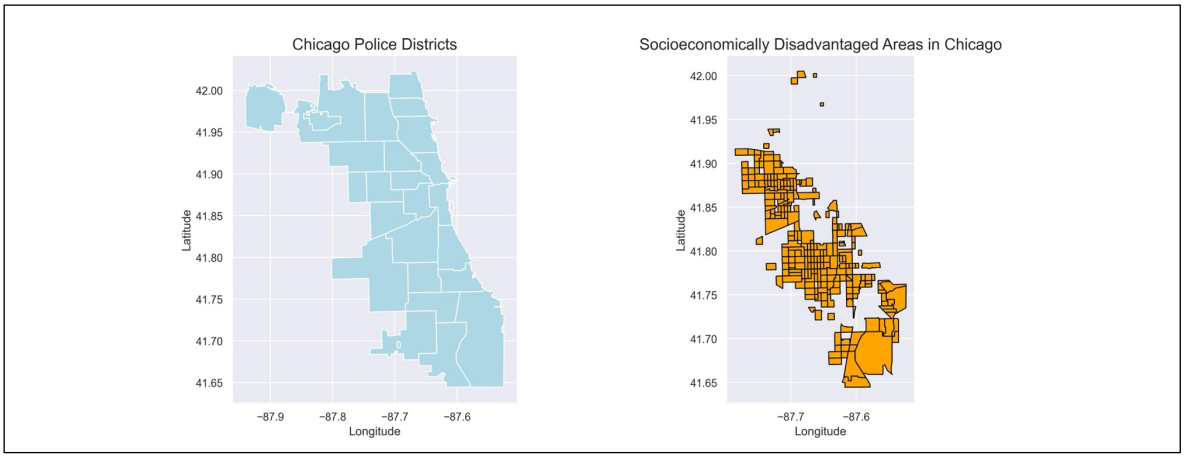

Common Crime Types
30k
20k
10k
0
37357
30022
26837
16195
12632
10152
8654
4924
3062
2568
2341
2080
2040
1680
1173
BATTERY
NARCOTICS
WEAPONS VIOLATION
THEFT
ASSAULT
CRIMINAL TRESPASS
OTHER OFFENSE
CRIMINAL DAMAGE
ERFERENCE WITH PUBLIC OFFICER
DECEPTIVE PRACTICE
PUBLIC PEACE VIOLATION
ROBBERY
MOTOR VEHICLE THEFT
BURGLARY
HOMICIDE
Crime Type
crime_count

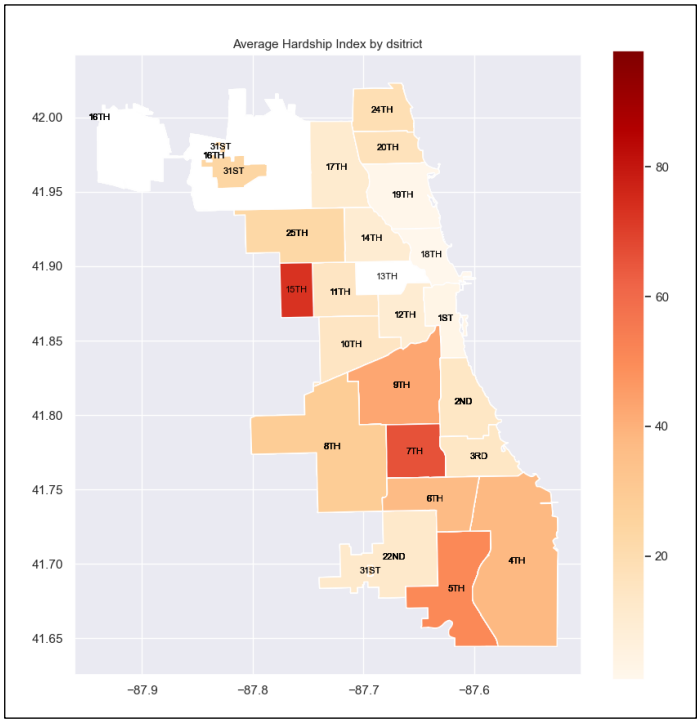

Arrest Counts by District
42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.9
−87.8
−87.7
−87.6
1400
1200
1000
8000
6000
4000
2000
16TH
31ST
24TH
20TH
17TH
19TH
25TH
14TH
18TH
15TH
11TH
13TH
12TH
1ST
10TH
9TH
2ND
8TH
7TH
3RD
6TH
22ND
4TH
5TH

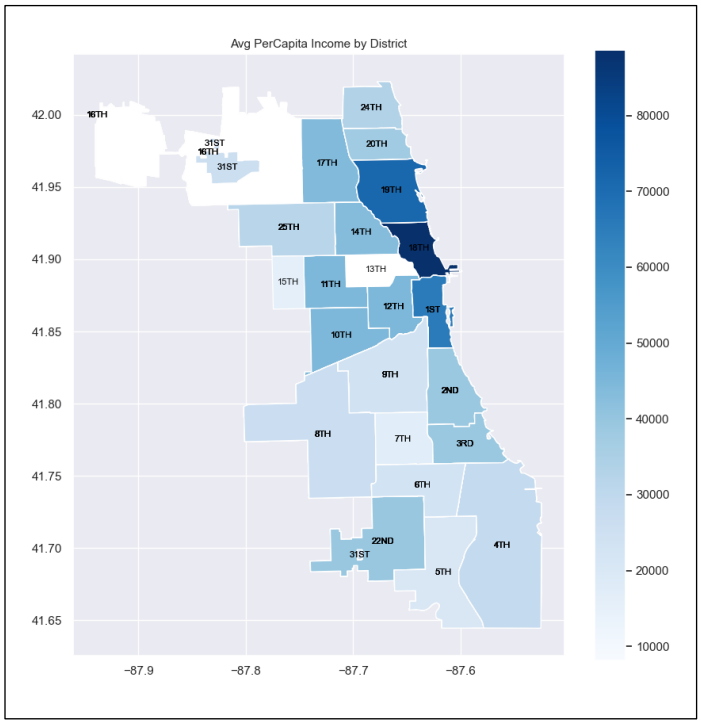

Population with age >25 without high school education
42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.9
−87.8
−87.7
−87.6
50
40
30
20
10
16TH
31ST
24TH
20TH
17TH
19TH
25TH
14TH
18TH
15TH
11TH
13TH
12TH
1ST
10TH
9TH
2ND
8TH
7TH
3RD
6TH
22ND
4TH
5TH


Heat Map of Crime Incidents by District
42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.9
−87.8
−87.7
−87.6
25000
20000
15000
10000
5000
16TH
31ST
24TH
20TH
17TH
19TH
25TH
14TH
18TH
15TH
11TH
13TH
12TH
1ST
10TH
9TH
2ND
8TH
7TH
3RD
6TH
22ND
4TH
5TH


Average percent households below poverty
42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.9
−87.8
−87.7
−87.6
35
30
25
20
15
16TH
31ST
24TH
20TH
17TH
19TH
25TH
14TH
18TH
15TH
11TH
13TH
12TH
1ST
10TH
9TH
2ND
8TH
7TH
3RD
6TH
22ND
4TH
5TH







4.4 Analysis Insights:

- Police District 11 in Chicago has highest rate of arrests.
- Most of the crime incidents happen on the street.
- Crime rates tend to rise towards weekend days of Friday, Saturday and Sunday.
- Crime rates seem higher towards march and April based on analysis of 5-year aggregated data.
- The most common crime types are Battery, narcotics, weapons violations and theft.
- Socio economic indicators seem to have a positive correlation with crime as socio-economically disadvantaged areas experience more crime.

4.5 Actions for call based on Analysis Insights:

- Pay more attention or allocate resources for specific police districts such as District 11 or on weekends.
- To prevent most crime incidents that happen on the street, police must act.
- Pay attention to most common types of crimes such as Battery, narcotics, weapons violations and theft.
- Pay attention to Socio economically disadvantaged areas to increase Per-Capita income and education level to proactively reduce crime rates.

5. Further objectives to improve the project

- Automate ingestion with automation tools like Azure data factory or Airflow.
- Automatic deployment of cloud using Infrastructure as a code (Terraform)
- Unit testing of code modules