NYC Crime Visualization: Final Report

Kuan-Lin Lee, Chan-Yu Cheng, Li-Yeh Yang, Mengqiao Cai

Introduction&Motivation

This project aims to visualize crime data in New York City by implementing a dashboard that displays information about crime types, time, date, and location. This information helps people identify high-crime areas and avoid them, increasing their safety. The dashboard allows users to select specific information, providing a more customized view of the data. This project has the potential to benefit the public and law enforcement agencies, by providing valuable insights into patterns and trends in criminal activity, allowing for more effective resource allocation. Additionally, the incorporation of this data into mapping systems could provide safer routes to users, potentially saving lives.

Project Outline

- Getting data from NYC OpenData
- Build a timeline of selected areas to understand the safety change overtime and histogram of different types of crime by seaborn.
- Interact with map by using gmplot drop pin on google map api for user to select area.
- Using ipywidgets to design UI. let users select which result they want.

Desired Code Functionality

- Read dataset from NYC OpenData: we can download crime information from NYC OpenData website. And reading this csv file is one of the functions our project has to achieve.
- Visualize dataset: we plan to visualize the dataset in two ways. One is to show the number between different types of crime. One is to show case

numbers change over time. We plan to visualize by matplotlib and seaborn tools.

- Data Analyst: From the data we show, we can know the safest place in New York, and the dangerous place we should avoid to attend. To do this we import sklearn to do k-cluster.
- Interact UI: to more easily understand, we decide to import google map UI to identify the information on the map.

Scientific background:

K-mean clustering

K-mean clustering is one of the unlabeled, unclassified data and enables the algorithm to operate on that data without supervision, which is more convenient for us to reach a wider audience. K-mean clustering only needs the k value and the algorithm can show the result we want.

Here is how the specific algorithm works.

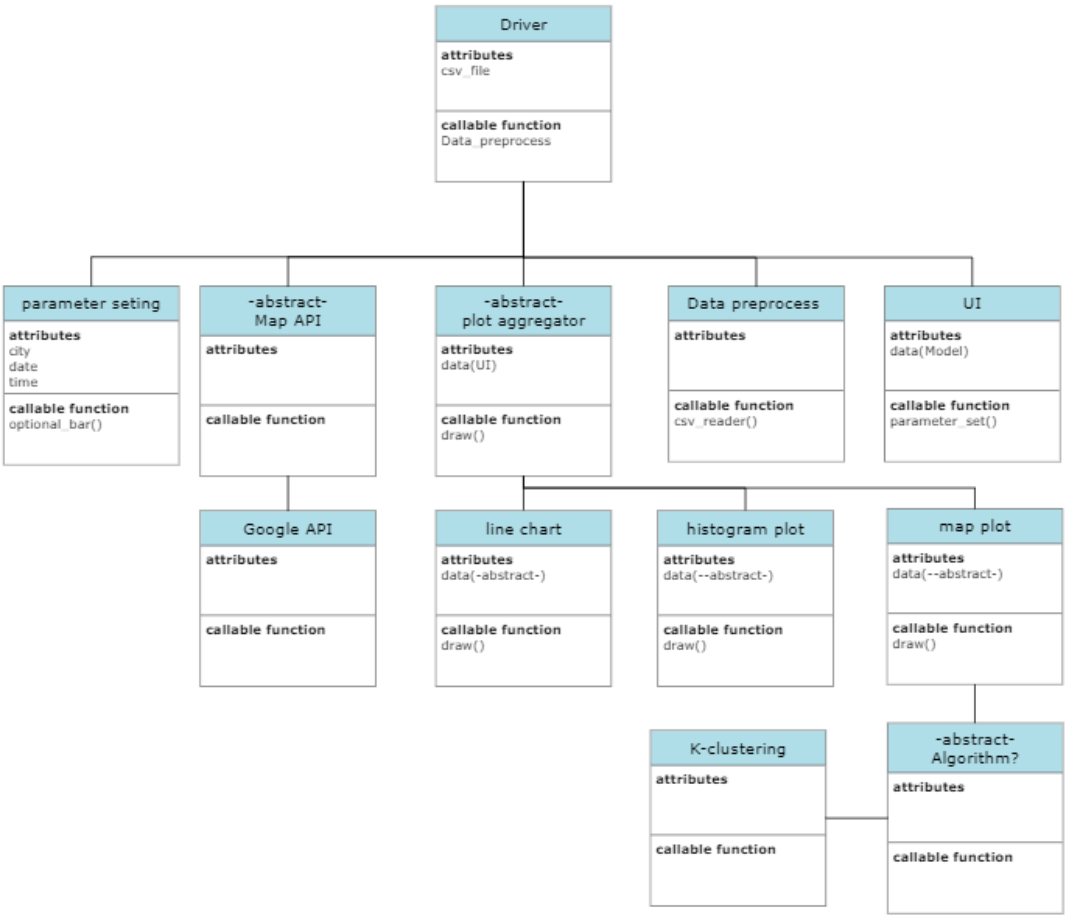
1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

UML diagram:



Project Goals:

In our project, we aim to showcase crime data in New York City using different visualization techniques. We plan to use a histogram to display crime types. The x-axis will represent the name of the crime, and the y-axis will show the number of crimes. We will also use a line chart to show the crime amount and time. The x-axis will represent time, and the y-axis will show the number of specific crimes during specific time sections. This will allow us to compare information between different time sections and show the crime amount during different years.

If we have enough time, we hope to integrate the crime data with Google Maps API. We plan to drop pins on the map to show users safe places more conveniently. With over 3 million data points, we plan to cluster the data to avoid cluttered information. We will use ipywidgets to build the dashboard, which will allow users to select specific information they want to see more

conveniently. We will use seaborn to draw the histogram and line chart, and incorporate gmplot to drop the pins on the map.

By using different visualization techniques, including the dashboard, histogram, line chart, and map, we believe we can present the crime data in a clear and concise manner. This project has the potential to benefit both the public and law enforcement agencies. The public can make informed decisions based on the data, while law enforcement can identify patterns and trends in criminal activity, allowing for more effective resource allocation. Additionally, the incorporation of this data into mapping systems could provide safer routes to users, potentially saving lives.

Dataset:

From the website NYC OpenData: https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Current-Year-To-Date-/5uac-w243, we could get the data on the crime in New York City. The data collected from 2021 until now and collected by NYPD. This dataset includes more than 360000 crimes and 24 different kinds of data on each crime. This dataset is keep updating. The website created by the New York Government (https://maps.nyc.gov/crime/) shows the result I am going to implement. But we cannot get the UI of this map, which means that we cannot use this map to make further development. The website I just mentioned is not open source, if we decided to add more features or different kinds of filters, it is impossible to do that. From the same website with this link (https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i), we can find the whole historic data, the data collected from 2016 until now, which includes more than 7.83 million crimes and 35 types.

Information visualize technique:

In our project, we aim to showcase crimes in New York City using various visualization techniques. We consider showing the crime types with a histogram, where the x-axis will display the name of the crime, and the y-axis will show the amount of each crime. For each specific crime, we also plan to display the crime amount and time with a line chart. Here, the x-axis will show the time, while the y-axis will represent the amount of a particular crime during a specific time section. By using the line chart, it will be easier to compare information between different time

sections, and it can also show the specific crime amount during different years. This way, we can determine whether the safety of a particular region is getting better or worse.

If we have enough time, we also plan to combine the crime data with Google Map API. Our goal is to drop pins on the map that will allow users to conveniently identify safe areas. Since we have more than 3000000 data points, we thought that doing clustering before dropping the pin would be a good idea to avoid messy data. We used ipywidgets to build the dashboard, which allows users to select the specific information they want to see more conveniently. For the histogram and line-chart, we used seaborn, and for drawing the pins on the map, we incorporated gmplot. By using the dashboard, histogram, line-chart, and map, we can show the visualization result clearly.

System Design:

For this project, we designed a system that takes in a large amount of crime data in CSV format, processes and visualizes it for user-friendly analysis. The system architecture consists of several components, including data cleaning and preparation, data visualization, and dashboard development. First, the data cleaning and preparation component cleans and processes the raw crime data, removes unnecessary information and formats it for visualization. Next, the data visualization component includes histogram and line chart visualization using the Seaborn library to provide a comprehensive view of the crime data. Finally, the dashboard development component uses the Jupyter Notebook platform, along with the ipywidgets library to create a user-friendly interface for data analysis. Additionally, we incorporated the Google Maps API to visualize the crime data on a map, which allows users to quickly identify high-crime areas and take necessary precautions. Overall, the system is designed to be modular, scalable, and user-friendly, enabling future expansion and customization.

Development Process:

We started the development process by defining the project scope and requirements. We conducted meetings with stakeholders to ensure we had a clear understanding of the data and what the visualization should achieve. After we had a solid understanding of the requirements, we moved

onto the data cleaning and pre-processing phase. This involved converting the raw crime data into a format that could be easily visualized and incorporated into the dashboard.

Next, we began designing the user interface and building the dashboard. We started with the histogram and line chart visualizations using the Seaborn library, and integrated them into the dashboard using ipywidgets. We then incorporated the Google Maps API, clustering the data to avoid clutter, and plotting the location pins on the map.

Once the core features were built, we conducted thorough testing to identify and resolve any bugs or issues. We also conducted user testing to gather feedback and refine the dashboard to improve the user experience.

Finally, we deployed the system, making sure it was hosted on a secure and reliable server, and provided ongoing maintenance and support to ensure the system remained functional and up-to-date. Throughout the development process, we adhered to industry best practices, including version control, code reviews, and testing, to ensure the system was robust and reliable.

Summary:

We focus on crimes in this project. The report of all the happened crimes is saved in a CSV file which is not visible enough to humans. Even if we extract the data, the data types are huge, and it is difficult for us to get only the information we need. And for humans, without a map, it is hard to know where a dangerous location is just by looking at the latitude and longitude. With our

visualization skills, we make the data visible and easy to understand. We have millions of data, so we need visualization tools to help us or users feel more comfortable with such a huge amount of data. By building a dashboard, users can select the information they want to view and remove unimportant data. The dashboard is also interactive, allowing users to select options or adjust thresholds to get the desired data easily.

With histograms and line charts, people can easily analyze and compare data between different periods. For example, they can compare crime numbers between different years or different time slots. By incorporating Google Maps, we make the data more user-friendly. We convert the latitude or longitude text data to real-world map data, so people can easily see where dangerous locations are and avoid them with our crime map.

Lessons Learned:

Through this project, we learned the importance of data visualization skills in making data more accessible and understandable to users. The initial raw data in a CSV file was not visible enough for humans, and the data types were vast and challenging to navigate. By implementing visualization techniques such as histograms, line charts, and maps, we were able to convert the data into a user-friendly format. We also found that incorporating interactive dashboards and tools allowed users to adjust thresholds and filter out unimportant data, making the data more personalized and useful for individual needs. Another crucial lesson was the importance of data clustering in handling large amounts of data. By clustering the data, we avoided messy and cluttered visualizations, which can be overwhelming and challenging to interpret. Overall, we learned that with proper visualization techniques and tools, data can be made accessible, understandable, and actionable for individuals and organizations.

Future Work:

In terms of future work, we can further improve the crime visualization project by adding more features and enhancing the existing ones. For example, we can incorporate machine learning techniques to predict potential crime hotspots based on past data, and display these predictions on the map. We can also add more filters to the dashboard to allow users to drill down into specific types of crimes or specific areas. In addition, we can integrate real-time crime data to keep the map

up to date with the latest crime information. Another possibility is to include demographic data to analyze how crime rates vary by age, gender, or ethnicity. Overall, there are many directions in which we can take this project to make it even more informative and useful for users.

References& external libraries used:

Ipywidgets
Seaborn
Pandas
Gmplot
Webbrowser
Os
Sklearn
matplotlib