

MASTER PROJECT  
FALL 2022

# Data Analysis and Visualization of Open data sets

Pritesh Ponnaboina  
Advisor: Dr. Stefan Robila  
Department of Computer Science  
Montclair State University  
Montclair, NJ 07043

## CONTENTS

LIST OF FIGURES: .....	3
LIST OF TABLES.....	4
ACKNOWLEDGEMENT.....	5
ABSTRACT:.....	6
INTRODUCTION: .....	7
LITERATURE REVIEW:.....	8
COMMERCIAL VEHICLE SPACE REQUIREMENTS .....	8
PARKING IN NEW YORK CITY .....	8
DOUBLE PARKING .....	10
ON-STREET PARKING AND LOADING .....	10
VEHICLE PARKING BEHAVIOR .....	13
PROJECT GOAL: .....	15
SYSTEM DESCRIPTION: .....	16
PROJECT DEVELOPMENT:.....	16
DATASET:.....	17
RESULT:.....	21
CONCLUSION: .....	36
BIBLOGRAPHY .....	37
GITHUB LINK: .....	38

## LIST OF FIGURES:

Figure 1 Graph of on street parking data [5].....	11
Figure 2 Available Parking Space [5] .....	12
Figure 3 Parking Behavior [5].....	13
Figure 4 Violations by Road Type [5] .....	14
Figure 5 Count for NY county violations .....	27
Figure 6 Type of violations and their count.....	29
Figure 7 Double Parking Violation code .....	30
Figure 8 NYC Map .....	32
Figure 9 Graph showing tickets in types of violations .....	35
Figure 10 Graph showing percentage for types of violations .....	36

# LIST OF TABLES

Table 1 Fines for Violations [4] .....	9
Table 2 Double Parking Violations .....	10
Table 3 Fields of Dataset [9] [6] .....	17
Table 4 NYC Open Dataset.....	18
Table 5 Dataset using Socrata API .....	23
Table 6 Dataset of selected fields .....	25
Table 7 Violations count with their license type .....	26
Table 8 County with their Precinct numbers .....	28
Table 9 plate with violation count output.....	30
Table 10 Double parking Violations.....	31
Table 11 Violation Percentage .....	33

# ACKNOWLEDGEMENT

I would like to thank Dr. Stefan Robila for taking the role as my MS project advisor and providing guidance throughout the project. Dr. Robila and I had weekly meetings and he upheld every commitment and was always accessible. He always encourages students in learning new technologies and progressing them for their future.

He encouraged me in taking this Big Data project and supported me at every point of time. Dr. Robila is very knowledgeable and offered suggestions and recommendations from the beginning of the project, starting from the topic and proposal to the very end, the final report and presentation. I think it's incredibly important to have a teacher and mentor who not only provides direction and inspiration but also challenge students, offers feedback and provides an engaging learning experience.

Thank you Dr. Robila for providing me with an experience and information that will benefit me and prepare me for these new career advancements in my life.

Thank you,  
Pritesh Ponnaboina

# ABSTRACT:

Big data is the most popular technology now-a-days in each sector like society, tech, health and finance sectors etc. To make accurate predictions and to effectively utilize large datasets. We must manipulate the dataset, clean it and visualize it in order to have a good understanding or identify new patterns. This project was intended to design and develop an interactive application representing 2022 New York City parking ticket violations.

In addition, the purpose of this project was to gain experience with advanced Python libraries. In this process a Google Collab notebook was used where we can build and run the code to get the desired outcomes, this can be utilized by any user to identify the existing violation tickets with a specific plate number at any time and extract those tickets based on the given query.

Big data analysis is often seen as a complex process for analyzing a large set of data to uncover hidden information and patterns. This information can then be used by different groups of people to make decisions, to run their businesses and processes more efficiently. Just as important as the analysis technique used, so in this way we present that information.

Data visualization is a pictorial or graphical format which is present to show the desired output data into a form easier to understand and highlighting outliers. Removing the noise from data and highlighting the useful information, enables decision makers to see analytics presented visually and internalize them quickly.

The main objective of this project is to familiarize myself with different tools and technologies used in data analysis, storing the data and displaying the data in a useful format. The scope of this project was to analyze a large data set and to develop a user interface.

The user interface should be able to allow users to extract the information that is useful and make it easier to extract the information for the users. This process is carried out by different tools and they are all managed by datasets.

# INTRODUCTION:

Our concept is to show the various types of violations and their data in the form of percentages that also allows easy visualization of vehicle violation tickets data from dataset. Using a graph, reflecting the predicted tickets through the dataset based on their various types of violated incidents. The browsing interface will help us represent the relation and correspondence between the tickets and number of violations.

In this project, I have preliminarily identified the data set and downloaded it with queries based on my requirement for small dataset, as I have taken only 5000 records from this dataset. I found the subset of the data and I have performed the operations to get the required outcome. After i queried the data, then i need to download it and give as an input in the Google Colab notebook [13] [1]. Google Colab notebook is used to execute the Python code with all required Python libraries. In the notebook, Python code along with all the libraries must be executed. We then need to print the given dataset file in order to process the data present in the dataset. I have plotted a graph by giving some inputs to the data, which I got after performing Python operations. Once we plot a graph, there will be a representation of our entire dataset of NYC traffic violations data. I would then be able to monitor trends in the data with respect to incidents that happened at a particular time.

We can also use Socrata API [4] instead of regular dataset [2]. As, Socrata API reduces the complexity of storing the dataset and retrieving the data multiple times. The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data. This API is very useful as the data would be easily processed.

## PROBLEM STATEMENT

The reason for assigning the tickets by the US government is not to make money but to change the behavior of the traffic violator. Many people are so conscious about money and when they violate any rule of traffic. Once they get a ticket, they will be worried, and they try to avoid getting tickets next time.

By checking the type of violation tickets percentage and by focusing on that aspect, we try to decrease the violations count in every violation type.

# LITERATURE REVIEW:

In the New York, For speeding ticket conviction the fines can range between \$90 - \$600. It will cost about \$45-\$150 for speeding over the limits. DMV is the Department of Motor vehicles for NY. DMV sets 3 points for speeding violation, which ranges in the levels of speed. In case, a person drives much more speed than the usual speeding violation, then he will be charged double for that. DMV has various rules to impose penalties for speeding tickets. Speeding tickets in school zone or work zone can cost twice as much as a regular speeding ticket. There are also particular codes known as DOF Parking Violation Codes.

## COMMERCIAL VEHICLE SPACE REQUIREMENTS

Today's market is continuously growing for direct to home deliveries of consumer goods and for commercial vehicle demand for off-street or on-street loading space in residential land use areas. Growth in residential demand and commercial freight, but the US government is unable to provide regulations for the accommodations of parking with growth in demand. Freight vehicle parking is one of the issues in urban areas. It has been observed that freight vehicles delivering different types of items parked for various periods of time. Whereas vehicles that are parked illegally are likely to park for a shorter period. This model is to estimate how much time the vehicles are illegally parking and also to study the parking durations of commercial vehicles.

## PARKING IN NEW YORK CITY

New York City has most residents and businesses located in North America with limited off-street parking. In Manhattan south of 66<sup>th</sup> street, there are parking spaces per unit and 1 space for each commercial development. This shows that the city requires more space for delivering goods.

Trucking firms are spending huge fines for their illegal vehicle parkings which costs approximately \$2000 per month. Usually parking fines add up to the revenue of NYC. In NYC, fines are based on violation type also on locations and times of day. Generally, Manhattan south 96<sup>th</sup> street has higher fines for parking violations.

There are different charges for each violation type based on the violation with respect to the location. When we do parking or standing in any no parking zone or no standing zone then we get fine of amount \$165. While we park the vehicle in that space, then we get \$65-515 based on the type of violation. For blocking the crosswalk or traffic lane or alongside curb then the fine would be \$115. Parking in a bus space around 7AM to 7PM timings then we would be fined \$115 in lower Manhattan area. Freight vehicles or any other vehicles loading or unloading in no standing zone will fine about \$95. If a person doesn't have any license documents or he is parking without proper registration, then they will be charged \$65 and will also be taken some



legal action. Parking meter violation happens when the vehicle is parked without paying at any paid parking zone or parked at any handicapped zone without an appropriate permit. If any person doesn't pay the fine in a stipulated time, then the judge may issue an FTA (Federal Transit Administration), an arrest warrant or they simply revoke the person's driving license. In some cases, the person might be jailed based on the circumstances. When we park our vehicles in a bus stop or bus lane then there will be a fine amount of up to \$115. For a double parking, the fine amount will be \$115.

General type	Specific violation	Location	Fine amount
Parking/Standing in a No Stopping/Standing/Parking Zone	Parking commercial vehicle overnight on residential street	Citywide	\$65–515
	Blocking sidewalk ramp	Citywide	\$165
	Parking/standing/stopping in intersection, crosswalk, traffic lane, no standing/stopping zone, or bicycle lane; alongside curb and blocking traffic; or on sidewalk	Citywide	\$115
	Parking in a bus space, 7 AM–7 PM	Lower Manhattan	\$115
	Loading/Unloading in no standing zone	Citywide	\$95
	Parking without proper registration/inspection/license documents	Citywide	\$65
Parking Meter Violations	Feeding meter, parking at broken/expired meter, not displaying parking meter receipt	Manhattan, south of 96th Street	\$65
		Elsewhere	\$35–65
Bus Stop/Lane	Parking in bus stop or bus lane	Citywide	\$115
Double Parking	When not allowed	Citywide	\$115

*Table 1 Fines for Violations [4]*

## DOUBLE PARKING

Parking is a vital component of urban areas. As demand for parking has increased over a certain time with land constraint, parking is becoming a major concern to urban areas. As demand for parking goes on increases, the situation will become worse in the future.

Double parking violations are a major concern of cities. It is defined as violation, when a person is standing or parking vehicle on the roadway side of a vehicle stopped or even parked at the curb. Double parking costs more amount than other types of violations. Total number of violations for passenger vehicles is higher than commercial vehicles but for double parking violations, commercial vehicle violations are higher.

When we compare the differences of double-parking behavior between commercial and passenger vehicles, we can point out that passenger vehicle is a one time violated type and commercial vehicle is multiple time violated types.

```
[ ] results_df.loc[results_df['violation']=="DOUBLE PARKING"]
```

	plate	state	license_type	summons_number	issue_date	violation_time	violation	fine_amount	penalty_amount	interest_amount	reduction_amount	payment_amount	amount_due	precinct	county
3	58797MK	NY	COM	8849324911	01/21/2021	08:26P	DOUBLE PARKING	115	0	0	80	35	0	108	Q
110	Y91MAF	NJ	PAS	8798083650	01/12/2021	10:52A	DOUBLE PARKING	115	0	0	0	115	0	066	K
144	JLF2157	NY	PAS	8889057919	02/20/2021	07:24A	DOUBLE PARKING	115	0	0	115	0	0	103	Q
208	EVG5774	NY	PAS	8849366176	12/04/2020	02:35P	DOUBLE PARKING	115	0	0	0	115	0	114	Q
215	JNK2532	NY	PAS	8884751044	03/27/2021	12:11P	DOUBLE PARKING	115	0	0	0	115	0	106	Q
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4719	T659055C	NY	OMT	8830326045	01/10/2022	04:41P	DOUBLE PARKING	115	0	0	0	115	0	103	Q
4761	LCX8112	PA	PAS	8982172695	09/27/2021	12:27P	DOUBLE PARKING	115	0	0	0	115	0	060	K
4776	JPV5179	NY	PAS	8891052176	01/05/2021	11:26A	DOUBLE PARKING	115	0	0	0	115	0	073	K
4899	87660KA	NY	COM	8823152124	02/22/2020	09:08A	DOUBLE PARKING	115	0	0	115	0	0	024	NY
4902	JH24440	NY	PAS	8891057563	01/11/2021	11:54A	DOUBLE PARKING	115	0	0	0	115	0	081	K

120 rows × 19 columns

*Table 2 Double Parking Violations*

## ON-STREET PARKING AND LOADING

For most of the reserved spaces, a signboard has been posted. Even curb spaces are provided with sign boards. Under each regulation, the space accommodations and the curb length is provided. There are regulations such as for Truck loading zone, No standing zone and like street cleaning days. Truck loading zone and No standing zones are available only for specific days. The available spaces are Open parking, Restricted parking, Commercial dedicated parking.

Open parking is the space where all types of vehicles park without any restriction and Restricted parking are spaces where all types of vehicles can park, but only for certain time. Commercial dedicated parking is the space where only commercial vehicles can park for loading or unloading purposes for a limited time.

Parking time periods are generally framed into 3 parts morning, midday, evening. Morning time ranges between 7AM-10AM and midday time ranges between 10AM-2PM and evening time ranges between 2PM-7PM.

The Figure below is about on-street parking that shows the total available parking space with each parking category in streets and avenues. Overall available parking is highest in residential and lowest in commercial areas. From the picture, we can estimate the analysis for the commercial areas in both avenues and street types of roads, both are occupied mostly by commercial dedicated vehicles. When we consider residential areas, in the streets there is plenty of parking available and in avenues there is plenty of restricted parking. That is because there are plenty of retail stores located which has most curb spaces are restricted because they tend to have a separate reserved parking.

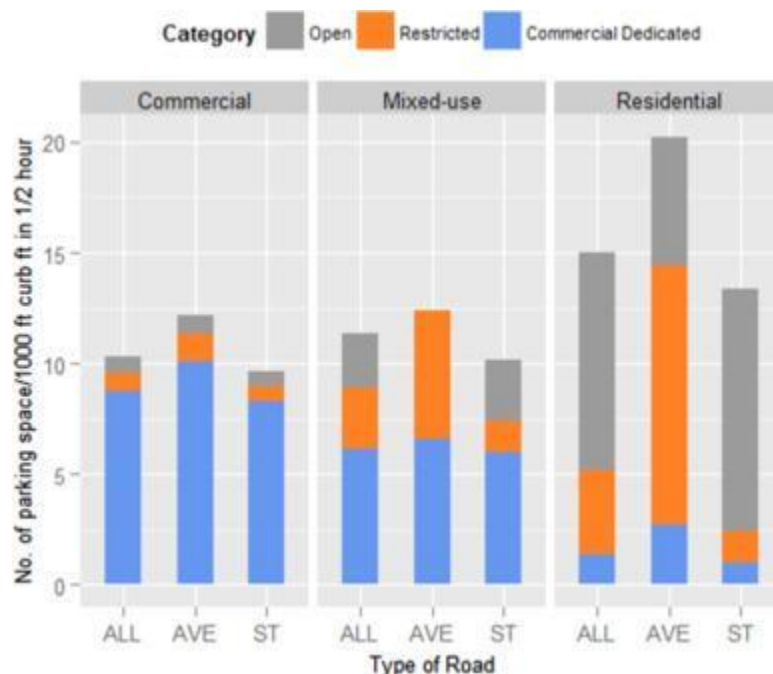


Figure 1 Graph of on street parking data [5]

This graph of Figure 1 shows that there is highest parking availability in residential areas. In commercial areas, most of the on-street parking on both types of roads is dedicated for commercial use.

In mixed-use areas, almost half of the area is covered for commercial use in both avenues and streets. While the other half is covered for time or meter restricted parking and unrestricted parking is also available in some streets. In residential areas, a completely different count is

observed. On avenues food establishments and retail stores are usually located, so most of the space is restricted and that is not dedicated for commercial use. On streets, where residents live park their personal vehicles, so most of the area is open for parking. There won't be any time restrictions except for some time during the weekly street cleaning. Also, on these streets parking rates are very low. So, commercial vehicles operators do not have access for legal parking on streets, and they usually park on avenues for their daily needs.

In residential census tracts, however, a very different pattern is observed. On avenues, where neighborhood retail stores and food establishments are generally located, most curb spaces (57%) are restricted, but are not dedicated for commercial use. On streets, where residents who do not have access to or do not wish to pay for off-street garages park their personal vehicles, 82% of parking is open, with no time or meter restrictions except for twice-weekly street cleaning that requires drivers to move vehicles for 30 min. On these streets, parking turnover rates are extremely low. As a result, commercial vehicle operators rarely have access to legal parking on streets, leading them to frequently park on avenues to serve nearby streets.

Figure below shows the parking space availability based on time of the day. In areas such as commercial spaces where parking spaces increase from morning till evening because those spaces are designed as 'No Standing' from morning to evening most probably from 8 am to 6pm and in between 10 am to 4 pm these spaces mostly parked by commercial vehicles.

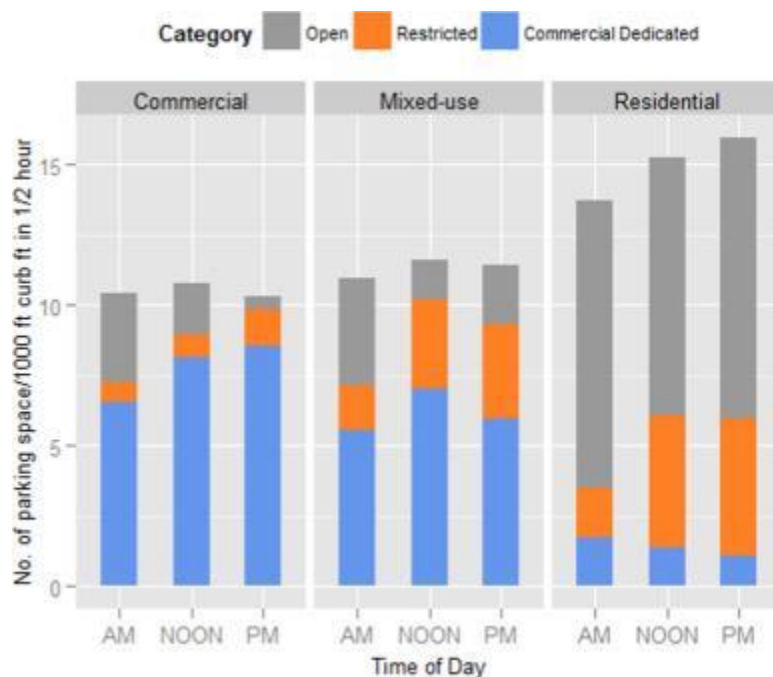


Figure 2 Available Parking Space [5]

By the above figure, we can estimate that when compared with mixed-use areas even commercial areas have the same statistics but when we look at overall mixed-use areas are slightly higher. In residential areas, restrictions will be there in the morning on street cleaning

which results in lowering overall parking availability. Some commercial activities happen in the morning and sometime in the midday periods. Basically, that would be around 7 am to 12 pm roughly.

## VEHICLE PARKING BEHAVIOR

To explore the various parking behaviors, violation records of all types are monitored. Parking violations are only captured when they violate the rules which result in illegal parking and are provided by fines. By curiously examining the types of violation and their frequency within each area type can show some differences in parking behavior. Violation rates can also be estimated by the sum of all violations divided by sum of available parking at a given time.

Most common violation types are categorized as 5 categories in total, they are No standing or parking, overtime parking, no meter receipt and double parking and other types.

'No standing or parking' category includes records from all the commercial vehicles parked in prohibited areas like bike lanes, no standing areas, pedestrian areas, bus stops etc.

'Overtime parking' is either a commercial vehicle's meter receipt expired or the vehicle which is parked in time restricted space for a longer time. 'Double parking' violation happens when parking is done in other than travel lane and it is also totally prohibited in Midtown Manhattan.

'No meter receipt' is when commercial vehicle occupied in legal metered space and meter is not even paid. 'Other' violations are like expired vehicle registration, and late inspection etc.

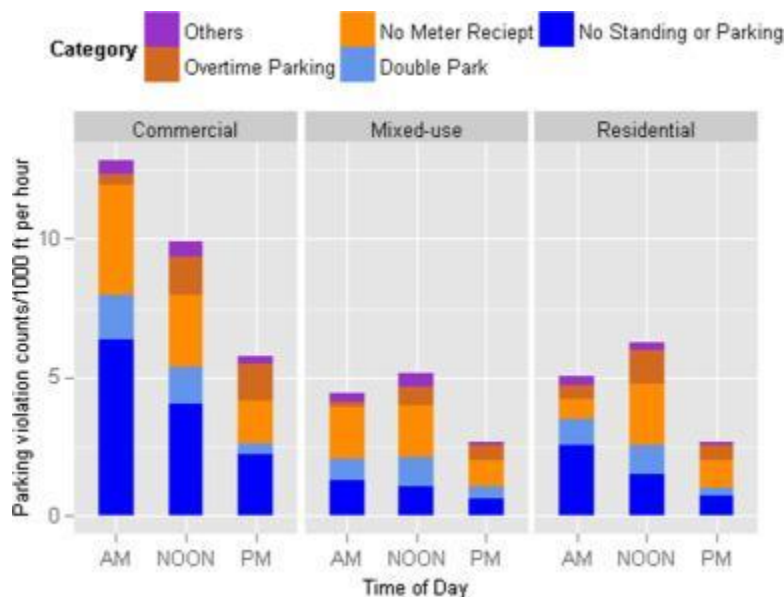


Figure 3 Parking Behavior [5]

The rate of parking violations in mixed-use and residential is little bit complicated than in commercial areas as several trips are included by both mixed-use and residential areas. The

rate in both areas is completely different from the commercial area graph because in commercial areas, there is a slight increase from Monday to midday and from midday to afternoon there is a much decrease. Anyhow, violation rates in commercial areas are higher during morning and during midday compared to those in mixed-use areas. As a result, we may know that there is a lower availability of commercial dedicated spaces in those areas. In residential areas there is high availability of facing 'No standing or parking' and Double parking' violations when compared to commercial areas that will count to 60% and in mixed-use areas it would be approximately 50% as we can see in the graph.

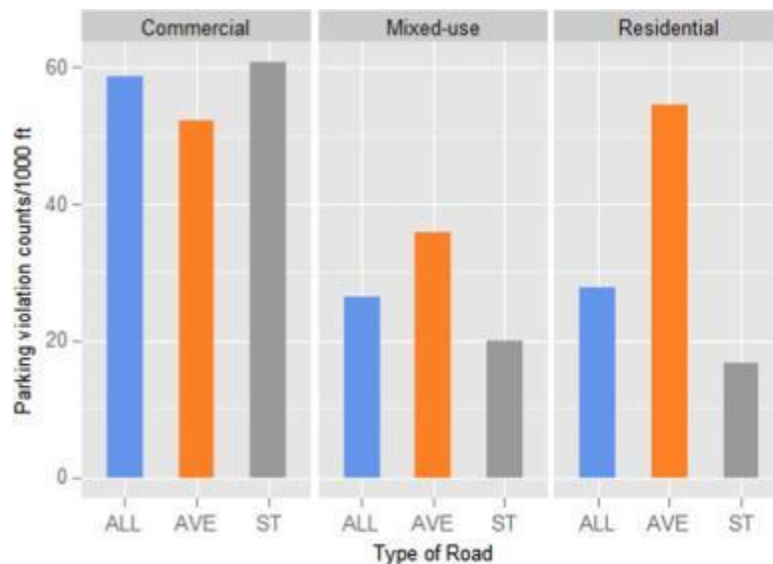


Figure 4 Violations by Road Type [5]

The Figure above shows violations by road type in each area type. We compare between avenue, street and all. Here all refers to both avenues and streets together. Now we compare the data in commercial, mixed-use and residential areas. In the commercial category, violations in streets are more than avenues. In the mixed-use category, avenues have higher violation rates than streets. In the residential category, the avenue violation rate is much higher than in the streets. When we compare all together, the violation rate in residential avenues are even higher than in commercial avenues. Commercial vehicles in residential areas can legally double park because of the single travel lane and with curbside parking on both sides. Usually, commercial vehicles tend to park on wider roads in avenues in which they can get better space whether it could be legal or illegal. Even though the commercial vehicles doesn't obstruct the road completely but this may interrupt high volume of vehicles, which results in time delays.

Moreover, some action must be taken as an alternative to this problem as commercial vehicle parking options for various road locations. This analysis is a broad estimate of the data and will also be useful to understand where the vehicles park and when in all these areas. This overall study relied on data which has been taken from secondary sources. Mostly relevant data cannot be accessed to common public. So, everything regarding this data is based on some research.

## PROJECT GOAL:

The goal of the project is to get the desired data of traffic violation tickets during some period from the complete dataset. While there is a flexibility of taking certain records from certain period. We collect all the data from all kinds of violation tickets and there are different prices for each violation ticket.

We must check for the number of tickets at a particular place, and we should also check whether the number of violations has increased or decreased. We can check for the fine amount for a particular vehicle based on its Plate number. Total number of tickets gives us the idea and using Python [4] [3] tools we can plot the graph.

Here we use Python functions to extract the required data from the server and filter it using functions.

In this python project, arranging the date field is useful as it helps to progress further in our project. We must progress more in pointing out the locations and mapping that location.

# SYSTEM DESCRIPTION:

## PROJECT DEVELOPMENT:

First, we need to install the pandas and matplotlib libraries, if you do not already have them. Approach to explore the most ticket violations across New York state. In our approach, we use several techniques to get open ticket violations. We use NYC Open data site to track all the violations and to download the csv file. We take the csv file, or even we can extend the JSON file and we direct it towards Socrata [9] for soda developers, which is a site for json objects. Once we have the csv file data, we can then view or visualize or export the data to any platform. There will also be Filters for filtering the data such as their violation time, data and license type and issue date. After filtering the data, we can check if there are any garbage values in the data, if we get then we can be able to filter that data using python in Jupyter notebook. One such technique to filter out the required data is to import the date fields in the python csv file classes. Convert to date file can be used to convert the date as per our requirements.

From the read csv file, we can filter the date format and can get the required output synchronously. We shall have to use the python methods to extract the data out. Once we perform this operation and we produce the results then we must visualize the graphs based on the data. We can then easily visualize complete data into a single picture. We can then get the finest set of required data [8] [4].

We have used several tools in this project like,

1. GitHub [3]– GitHub is a popular source code maintenance tool. It integrates well with IDEs such as Visual Studio Code and Google Colab to provide code versioning and code deployment.

2. GitHub [3]is the largest open-source community and provides the creation of code repositories.

GitHub desktop is included with the account, and it enables a developer to maintain a local repository and commit changes to the cloud as and when needed.

3. Socrata API [4] and SoQL – NYC Open Data web site recommends Socrata API [3 [5]] to be used for

data extraction or data interaction. Socrata API provides “Socrata Query Language” – SoQL – for

using the API to query the data from the data source. Socrata provides 3 ways to access the open data – CSV, JSON and API end point. A developer can access API end points to either

download the full data set or query a subset of data by appending SoQL queries to the API endpoints.

4. Google Colab [1] or any other IDE – Google Colaboratory is a Data Science IDE by Google to develop deep learning models. This is a cloud hosted IDE and provides GPU support with RAM around 12GB to train machine learning models. The IDE infrastructure is supported by Google and it offers free account creation. This tool provides source code integration with GitHub and Google Drive. Developers can import their code directly from GitHub and perform



version control from within the Google Colab. This IDE also allows easy sharing of the source code with anyone having a Gmail account.

## DATASET:

Each type of data field has been taken from the Socrata dataset([data.cityofnewyork.us](https://data.cityofnewyork.us)). All the data fields have different data types like text and number. All the data fields have the capacity to return the records simply by using Socrata queries. Here, plate data field returns textual data, state data field also returns text data, license\_type data field also returns text data, summons\_number returns data in numbers. issue\_date, violation\_time, violation, judgement\_entry\_date all has same text data also precinct, county, issuing\_agency, violation\_status all has text data. fine\_amount, penalty\_amount, interest\_amount, reduction\_amount, payment\_amount, amount\_due. summons\_image returns url as an output.

⊕	plate	text	Plate
⊕	state	text	State
⊕	license_type	text	License Type
⊕	summons_number	number	Summons Number
⊕	issue_date	text	Issue Date
⊕	violation_time	text	Violation Time
⊕	violation	text	Violation
⊕	judgment_entry_date	text	Judgment Entry Date
⊕	fine_amount	number	Fine Amount
⊕	penalty_amount	number	Penalty Amount
⊕	interest_amount	number	Interest Amount
⊕	reduction_amount	number	Reduction Amount
⊕	payment_amount	number	Payment Amount
⊕	amount_due	number	Amount Due
⊕	precinct	text	Precinct
⊕	county	text	County
⊕	issuing_agency	text	Issuing Agency
⊕	violation_status	text	Violation Status
⊕	summons_image	url	Summons Image

Table 3 Fields of Dataset [9] [6]

NYC traffic violations open dataset consists of all the fields like Plate, State, License Type, Summons Number, Issue Date, Violation Time, Violation, Judgement Entry Date, Fine Amount,

Penalty Amount, Interest Amount, Reduction Amount, Payment Amount, Amount Due, Precinct, County, Issuing Agency, Violation Status, Summons Image.

Plate number acts as a primary identifier for any vehicle when it is violated. The state can only be NY because we are only taking New York traffic violations data. Summons Number will be a 10-digit number given for all the vehicle holders and Issue date and violation time will be the date when the vehicle is violated in traffic. The violation field shows the type of violation. Fine amount shows the total fine incurred for the following violation. Penalty amount shall be added if some person delays the payment for the previously occurred violation. An interest amount will be added only if the case is entered into judicial court which takes about 100 days from the day of violation. The payment amount is the actual amount which must be paid. If there are any dues, then the amount will be shown in the amount due column.

Precinct number is the area around the building, it is used to identify the nearest area location and used along with the address. County is an administrative or political subdivision of a state that consists of a geographic region with specific boundaries. Issuing agency is the one who took that ticket.

In order to access this dataset, you must visit <https://data.cityofnewyork.us/City-Government/Open-Parking-and-Camera-Violations/nc67-uf89> [8][2].

This CSV file will be too large to download(nearly 13gb), so you can query the data using the filters present in the dataset and get the desired length of records to make some operations using the dataset.

	Plate	State	License Type	Summons Number	Issue Date	Violation Time	Violation	Judgment Entry Date	Fine Amount	Penalty Amount	Interest Amount	Reduction Amount	Payment Amount	Amount Due	Precinct	County	Issuing Agency	Viol Status
0	7BIGCAT	NJ	PAS	8825438758	09/06/2022	10:50A	FAIL TO DSPLY MUNI METER RECPT	NaN	35.0	0.0	0.0	0.0	35.0	0.0	26.0	NY	TRAFFIC	
1	JJB4730	NY	PAS	8825642039	08/31/2022	10:15A	FIRE HYDRANT	NaN	115.0	0.0	0.0	0.0	115.0	0.0	115.0	Q	TRAFFIC	
2	KPP9008	NY	PAS	8842274604	09/02/2022	08:18A	NO PARKING-STREET CLEANING	NaN	65.0	0.0	0.0	0.0	65.0	0.0	30.0	NY	TRAFFIC	
3	T756158C	NY	OMT	8842275141	09/03/2022	09:37A	NO PARKING-DAY/TIME LIMITS	NaN	60.0	0.0	0.0	0.0	60.0	0.0	30.0	NY	TRAFFIC	
4	KMUJ2303	NY	PAS	8842275219	09/06/2022	06:12A	NO PARKING-DAY/TIME LIMITS	NaN	60.0	0.0	0.0	0.0	60.0	0.0	28.0	NY	TRAFFIC	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12461556	KHL6422	NY	OMS	8824848503	09/06/2022	09:09A	NO PARKING-STREET CLEANING	NaN	65.0	0.0	0.0	0.0	65.0	0.0	78.0	K	TRAFFIC	
12461557	HXX3843	NY	PAS	8824848588	09/06/2022	09:26A	NO PARKING-STREET CLEANING	NaN	65.0	0.0	0.0	0.0	65.0	0.0	78.0	K	TRAFFIC	

Table 4 NYC Open Dataset

The above picture shows the traffic violation data.

Violation status and Summons image fields are at the right corner of the picture.

I used several python libraries to extract the data and perform some operations on the data in this dataset such as

1. `pip install geopandas`  
`import geopandas as gpd`
  - This Python library is used to work with geospatial data in Python easier. Geopandas extend the datatypes used by pandas [7].
2. `pip install contextily`  
`import contextily`
  - It is a small package to retrieve and write to disk tile maps from the internet into geospatial raster files [8].
3. `pip install folium`  
`import folium`
  - Folium is a powerful Python library that helps you create several types of Leaflet maps [8].
4. `pip install matplotlib`, `pip install matplotlib.pyplot as plt`  
`import matplotlib`, `import matplotlib.pyplot as plt`
  - This is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy [9].
5. `pip install mapclassify`  
`import mapclassify`
  - Mapclassify is an open-source Python library Choropleth map classification [10].
6. `pip install pandas`  
`import pandas`
  - It is a Python library for data analysis and manipulation tool [11].
7. `import os`  
`os.getcwd()`
  - OS module in Python provides functions for creating and removing a directory, fetching its contents, changing and identifying the current directory.
8. `pip install geopy`  
`import geopy`
  - Geopy is a Python client for several popular geocoding web services. Geopy makes it easy for Python developers to locate the coordinates of address [8].
9. `pip install sodapy`  
`import sodapy`
  - It is a Python client for the Socrata Open Data API [8].
10. `pip install geoplot`  
`import geoplot`
  - It is a high-level Python geospatial plotting library. It's an extension to cartopy and matplotlib which makes mapping easy [8].

11. pip install numpy as np

```
import numpy as np
```

- It is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform [12].

12. pip install seaborn as sns

```
import seaborn as sns
```

- It is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics [9].

13. pip install opendatasets

```
import opendatasets
```

- It is a Python library for downloading datasets from online sources like google drive [4].

14. pip install geoviews

```
import geoviews
```

- Geoviews is a Python library that makes it easy to explore and visualize geographical, meteorological and oceanographic datasets. [8]

I used all these python libraries for performing various operations and to get the desired outcome. I have used NYC traffic violations dataset and Soda API for output flexibility and to avoid delay outcomes.

# RESULT:

## STEP 1: Understanding the dataset

We are considering JSON format data from API and converted into python list of dictionaries by sodapy. In this project, we are utilizing two python packages pandas and sodapy. With pandas, the data will be retrieved and put into pandas data frame. The sodapy is used to retrieve and analyze data from Socrata open data sources without downloading the data, which will save the time and memory space.

Unauthenticated Clients on Socrata data, works only on public datasets without the need of username and password. Whereas Authenticated clients works for non-public datasets only where username and password will be needed.

Data taken from the API by using the client base and filtering out the data with 5000 records. This can be done by giving input as 'limit=5000'. After setting the limit to 5000, we can get 5000 records only from the dataset. After that, we define that result to pandas dataframe.

## STEP 2: Reading or Importing the Data

In this method, we don't use CSV files. As CSV file takes more time for the functioning and also some storage issues. So, we can also consider the same dataset using API.

When we consider the whole dataset using API, we can filter out and consider the small portion of required data. Here I have taken the data of 5000 sample records of traffic violation.

```
#https://dev.socrata.com/foundry/data.cityofnewyork.us/nc67-uf89
```

```
import pandas as pd
from sodapy import Socrata
```

```
# Unauthenticated client only works with public data sets. Note 'None'
# in place of application token, and no username or password:
client = Socrata("data.cityofnewyork.us", None)
```

Here we define password as None.

```
#APIKey ID: 8snpjf5l16qfu634ab2d2tvih
#APIKey: 56k8kcn3t2qint7xcl4cwn2owglxboncxioa19p8fey1ln3hol
```

```
# Example authenticated client (needed for non-public datasets):
```

```
# client = Socrata(data.cityofnewyork.us,  
#                 MyAppToken,  
#                 username="user@example.com",  
#                 password="AFakePassword")  
  
# First 5000 results, returned as JSON from API / converted to Python list of  
# Dictionaries by sodapy.  
results = client.get("nc67-uf89", limit=5000)  
  
# Convert to pandas DataFrame  
results_df = pd.DataFrame.from_records(results)
```

Here, I have considered the whole data frame in pandas. In order to do several tasks in data analysis, we must consider the data frame in pandas.

Once, we have that data, we then print the whole data using,

```
print(results_df)
```

```

print(results_df)

```

	plate	state	license_type	summons_number	issue_date	violation_time	\
0	61713MG	NY	COM	8974257257	04/26/2021	09:54A	
1	JNW4714	NY	PAS	8887702810	12/31/2020	08:56A	
2	8KXZ837	CA	PAS	8974256939	04/21/2021	09:27A	
3	JRC2415	NY	PAS	8871433282	07/24/2021	07:43A	
4	C124523	CT	999	8974257373	04/27/2021	08:40A	
...	...	...	...	...	...	...	
4995	KLT6923	NY	PAS	1484728890	10/09/2021	10:05A	
4996	35168TB	VA	PAS	1485050730	09/27/2021	11:15A	
4997	HRH7933	NY	PAS	1485090805	09/29/2021	07:46P	
4998	LDY2616	PA	PAS	2003078319	08/13/2021	09:10A	
4999	BXLYN	NY	ORG	8819148596	10/28/2020	09:38A	

	violation	fine_amount	penalty_amount	\
0	NO PARKING-STREET CLEANING	65	0	
1	NO PARKING-STREET CLEANING	65	0	
2	FAIL TO DSPLY MUNI METER RECPT	35	0	
3	NO PARKING-STREET CLEANING	65	0	
4	NO PARKING-STREET CLEANING	65	0	
...	...	...	...	
4995	NON-COMPLIANCE W/ POSTED SIGN	60	0	
4996	PEDESTRIAN RAMP	165	0	
4997	NO STANDING-EXC. AUTH. VEHICLE	95	0	
4998	NO PARKING-STREET CLEANING	65	0	
4999	FRONT OR BACK PLATE MISSING	65	0	

	interest_amount	reduction_amount	payment_amount	amount_due	precinct	\
0	0	40	25	0	062	
1	0	0	65	0	067	
2	0	0	35	0	076	
3	0	0	65	0	110	
4	0	65	0	0	070	
...	...	...	...	...	...	
4995	0	0	60	0	084	
4996	0	0	165	0	013	
4997	0	0	95	0	052	
4998	0	0	65	0	073	
4999	0	0	65	0	106	

	county	issuing_agency	violation_status	\
0	K	TRAFFIC	HEARING HELD-GUILTY REDUCTION	
1	K	TRAFFIC	NaN	
2	K	TRAFFIC	NaN	
3	Q	TRAFFIC	NaN	
4	K	TRAFFIC	HEARING HELD-NOT GUILTY	
...	...	...	...	

Table 5 Dataset using Socrata API

plate, state, license\_type, summons\_number, issue\_date, violation\_time, violation, fine\_amount, penalty\_amount, interest\_amount, reduction\_amount, payment\_amount, amount\_due, precinct, county, issuing\_agency, violation\_status, summons\_image, judgement\_entry\_date are the fields of the table in this dataset.

## STEP 3 : Cleaning the Data

Once we import the dataset, we must clean the data by finding the missing values and dropping the columns which are no longer needed. Here we are filtering the data to get the appropriate data. Here we are only considering the columns which have complete data. We can then use this copy of data for our further part of the data analytics project.

### results\_df.info()

This will print the data and their datatype.

```
cols_to_keep = ['state', 'plate', 'license_type', 'issue_date', 'violation_time',  
'violation', 'fine_amount', 'penalty_amount', 'interest_amount', 'precinct',  
'county', 'violation_status', 'summons_image', 'issuing_agency']
```

We can decide which fields to be inserted into our table by simply selecting all the required fields and executing the following code. I have also selected 5000 records only, so only 5000 records will only be shown as 5000 rows and 14 columns as we have selected few columns of data fields.

The code to generate the following table is

```
df = results_df[cols_to_keep].copy() # making a copy detaches this from the original sa  
mple_df and raw_df  
df
```

This code gives the output as the copy of the dataframe. The selected fields from the cols\_to\_keep will be only displayed in the dataframe, all other columns are ignored.



	state	plate	license_type	issue_date	violation_time	violation	fine_amount	penalty_amount	interest_amount	precinct	county	violation_status	summons_image	issu
0	NY	JEJ4178	PAS	08/24/2021	09:22A	FIRE HYDRANT	115	10	0	048	BX	NaN	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
1	NY	GBW1347	PAS	08/31/2021	01:06P	NO STANDING-DAY/TIME LIMITS	115	0	0	044	BX	HEARING HELD-GUILTY	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
2	NY	JPY5376	PAS	05/20/2021	03:25P	FAIL TO DSPLY MUNI METER RECPT	35	0	0	107	Q	NaN	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
3	NY	98578JW	COM	01/25/2022	08:57A	NO STANDING-EXC. TRUCK LOADING	95	0	0	001	NY	HEARING HELD-NOT GUILTY	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
4	NY	HNE5868	PAS	05/21/2021	08:53A	NO PARKING-STREET CLEANING	65	0	0	112	Q	NaN	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	NY	KSP9082	PAS	05/27/2022	07:29A	NO PARKING-STREET CLEANING	65	60	2.25	073	Kings	NaN	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url': DEF OF S/
4996	NY	32719MM	COM	05/20/2022	07:40A	NO STANDING-COMM METER ZONE	115	0	0	014	NY	HEARING HELD-NOT GUILTY	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':
4997	NY	H11239	PAS	08/03/2021	03:26P	FAIL TO DSPLY MUNI METER RECPT	35	0	0	120	R	NaN	'http://nycserv.nyc.gov/NYCServWeb/Sho...	{'url':

Table 6 Dataset of selected fields

## STEP 4: Modeling the Data

```
results_df.groupby('violation')['license_type'].apply(lambda x: x.value_counts().head(1)).
reset_index(name='Count').rename(columns={'level_1': 'license_type'})
```

The above code yields the output as total violations with respect to its license\_type also their count. As we can see the total violations with their type of license and their total count of their combinations below.

	violation	license_type	Count
0	ANGLE PARKING	PAS	3
1	BEYOND MARKED SPACE	PAS	3
2	BIKE LANE	PAS	6
3	BUS LANE VIOLATION	OMT	4
4	COMML PLATES-UNALTERED VEHICLE	COM	3
5	CROSSWALK	PAS	25
6	DETACHED TRAILER	PAS	12
7	DIVIDED HIGHWAY	PAS	2
8	DOUBLE PARKING	PAS	87
9	DOUBLE PARKING-MIDTOWN COMML	COM	8
10	EXPIRED MUNI METER	PAS	92
11	EXPIRED MUNI MTR-COMM MTR ZN	COM	12
12	FAIL TO DISP. MUNI METER RECPT	COM	33
13	FAIL TO DSPLY MUNI METER RECPT	PAS	265
14	FAILURE TO STOP AT RED LIGHT	PAS	618
15	FEEDING METER	PAS	1
16	FIRE HYDRANT	PAS	228
17	FRONT OR BACK PLATE MISSING	PAS	85

Table 7 Violations count with their license type

If we want to find the day of the week then we can use the code below and check for the issue date, license type and its county

```

week_dict = {0: 'Mon', 1: 'Tue', 2: 'Wed', 3: 'Thu', 4: 'Fri', 5: 'Sat', 6: 'Sun'}
weekly_df = results_df[['issue_date', 'license_type', 'county']]
weekly_df['dow'] = weekly_df['issue_date']
results_df = weekly_df.groupby(['dow'])['issue_date'].count().reset_index()
weekly_df['dow'] = weekly_df['dow'].map(week_dict)
fig = plt.figure(figsize = (10, 5))

x = weekly_df['dow']

```

```

y = weekly_df['issue_date']
plt.bar(x, y, color = purple, width = 0.8)
plt.xlabel("Day of week")
plt.ylabel("Number of parking Violations")
plt.title("Number of parking violations across days of the week", fontsize
=20)
plt.show()

```

To find out the violations in particular county (NY), then we can frame the code and the output given below

```

results_df[results_df['county'] == 'NY'].groupby('violation')['issue_date'].count().nlargest(5).reset_index(name='Count')

```

	violation	Count
0	NO PARKING-STREET CLEANING	161
1	NO STANDING-DAY/TIME LIMITS	96
2	NO PARKING-DAY/TIME LIMITS	60
3	FAIL TO DISP. MUNI METER RECPT	53
4	NO STANDING-COMM METER ZONE	44

Figure 5 Count for NY county violations

Here, in the above figure, we can observe that NY county has the highest violations rate in 'NO PARKING-STREET CLEANING' type of violations.

Precinct numbers are a part of territory with definite bounds or functions often established for administrative purposes such as a subdivision of a county, town or city for election purposes.

'Manhattan': list(range(1,35)),  
 'Bronx': list(range(40,53)),  
 'Brooklyn': list(range(60,95)),  
 'Queens': list(range(100,116)),  
 'Staten Island': list(range(120,124))

For every county, there are fixed precinct numbers. For the first 35 precinct numbers the county will be Manhattan and the list continues till 124.

In this we can also rename the names of the county as 'K' for Brooklyn, Q for Queens, NY for Manhattan, R for Staten Island, BX for Bronx, KINGS for Brooklyn.

To find out the violations and their precinct numbers, we use this code.


```

results_df = results_df[['issue_date', 'county', 'violation', 'precinct']].copy()

```

results\_df

This will produce the output as



	issue_date	county	violation	precinct
0	12/04/2014	Q	INSP. STICKER-EXPIRED/MISSING	105
1	11/18/2014	Q	INSP. STICKER-EXPIRED/MISSING	105
2	07/12/2021	BK	PHTO SCHOOL ZN SPEED VIOLATION	000
3	05/07/2019	K	NO PARKING-STREET CLEANING	070
4	04/11/2019	K	FAIL TO DSPLY MUNI METER RECPT	068
...	...	...	...	...
4995	12/18/2014	K	NO PARKING-STREET CLEANING	083
4996	04/12/2019	NY	FAIL TO DISP. MUNI METER RECPT	018
4997	11/22/2021	BK	BUS LANE VIOLATION	000
4998	09/02/2021	QN	PHTO SCHOOL ZN SPEED VIOLATION	000
4999	09/02/2021	BX	PHTO SCHOOL ZN SPEED VIOLATION	000

5000 rows × 4 columns

*Table 8 County with their Precinct numbers*

We can also find the count on Type of violations with the below code represented as,

```
results_df.violation.value_counts()
```

Output is displayed as,

PHTO SCHOOL ZN SPEED VIOLATION	1123
FAIL TO DSPLY MUNI METER RECPT	540
NO PARKING-STREET CLEANING	494
NO PARKING-DAY/TIME LIMITS	363
NO STANDING-DAY/TIME LIMITS	324
EXPIRED MUNI METER	220
FAILURE TO STOP AT RED LIGHT	209
INSP. STICKER-EXPIRED/MISSING	182
DOUBLE PARKING	154
FIRE HYDRANT	152
REG. STICKER-EXPIRED/MISSING	126
BUS LANE VIOLATION	117
NO STANDING-BUS STOP	110
NO STANDING-EXC. TRUCK LOADING	92
FRONT OR BACK PLATE MISSING	76
NO STANDING-EXC. AUTH. VEHICLE	54
FAIL TO DISP. MUNI METER RECPT	48
NO STANDING-COMM METER ZONE	41
CROSSWALK	37
BIKE LANE	30
PLTFRM LFTS LWRD POS COMM VEH	29
DOUBLE PARKING-MIDTOWN COMM	24
NO STOPPING-DAY/TIME LIMITS	23
EXPIRED MUNI MTR-COMM MTR ZN	18
SIDEWALK	17
COMM PLATES-UNALTERED VEHICLE	15
NO STANDING-TAXI STAND	15
INSP STICKER-MUTILATED/C'FEIT	14
NO STANDING-BUS LANE	14
NGHT PKG ON RESID STR-COMM VEH	12

*Figure 6 Type of violations and their count*

This figure depicts the total violations for a certain period. When we consider 5000 records sample, the most common parking violation is said to be PHTO SCHOOL ZN SPEED VIOLATION with most count will be said as the common violation. Then followed by FAIL TO DISPLAY MUNI METER RECPT and NO PARKING-STREET CLEANING and NO PARKING-DAY/TIME LIMITS and so on.

We can also focus on plates that are having most of the violation type tickets. Here, we can check the code for this scenario.

```
results_df.groupby('violation')['plate'].apply(lambda x: x.value_counts().head(1)).reset_index(name='Count').rename(columns = {'level_1': 'plate'})
```

The output for the following code will be displayed as,

	violation	plate	Count
0	ANGLE PARKING	ZE3K1X	1
1	ANGLE PARKING-COMM VEHICLE	71280MD	1
2	BEYOND MARKED SPACE	Z74HUL	1
3	BIKE LANE	69643T3	2
4	COMML PLATES-UNALTERED VEHICLE	XC115C	1
5	CROSSWALK	581807R	1
6	DETACHED TRAILER	TTU76T	3
7	DIVIDED HIGHWAY	JGG2621	1
8	DOUBLE PARKING	31762MN	4
9	DOUBLE PARKING-MIDTOWN COMML	22030MK	3
10	EXPIRED MUNI METER	XBYW44	3
11	EXPIRED MUNI MTR-COMM MTR ZN	10081NA	1
12	FAIL TO DISP. MUNI METER RECPT	PKV1850	2
13	FAIL TO DSPLY MUNI METER RECPT	AP20967	3
14	FEEDING METER	HTW9904	1
15	FIRE HYDRANT	89752P9	3
16	FRONT OR BACK PLATE MISSING	V42LCL	6
17	IDLING	8DK8631	1
18	INSP STICKER-MUTILATED/C'FEIT	JLU7895	1
19	INSP. STICKER-EXPIRED/MISSING	88241MK	3
20	MISSING EQUIPMENT	GXBT12	1
21	NGHT PKG ON RESID STR-COMM VEH	32537NA	1
22	NO MATCH-PLATE/STICKER	JJZ4602	2
23	NO PARKING-DAY/TIME LIMITS	ETX595	5
24	NO PARKING-EXC. HOTEL LOADING	94287MK	1
25	NO PARKING-STREET CLEANING	EZS7478	2
26	NO STANDING-BUS LANE	2342150	2
27	NO STANDING-BUS STOP	TN02269	2

Table 9 plate with violation count output

In this project, as we are also focusing on Double parking violations. I have also tried to find Double Parking violations and used this code.

```
] results_df.loc[results_df['violation']=="DOUBLE PARKING"]
```

Figure 7 Double Parking Violation code

This code produces the output as

	plate	state	license_type	summons_number	issue_date	violation_time	violation	fine_amount	penalty_amount	interest_amount	reduction_amount	payment_amount	amount_due	precinct	county	issu
3	58797MK	NY	COM	8849324911	01/21/2021	08:26P	DOUBLE PARKING	115	0	0	80	35	0	108	Q	
110	Y91MAF	NJ	PAS	8798083650	01/12/2021	10:52A	DOUBLE PARKING	115	0	0	0	115	0	066	K	
144	JLF2157	NY	PAS	8889057919	02/20/2021	07:24A	DOUBLE PARKING	115	0	0	115	0	0	103	Q	
208	EVG5774	NY	PAS	8849366176	12/04/2020	02:35P	DOUBLE PARKING	115	0	0	0	115	0	114	Q	
215	JNK2532	NY	PAS	8884751044	03/27/2021	12:11P	DOUBLE PARKING	115	0	0	0	115	0	106	Q	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4719	T659055C	NY	OMT	8830326045	01/10/2022	04:41P	DOUBLE PARKING	115	0	0	0	115	0	103	Q	
4761	LCX8112	PA	PAS	8962172695	09/27/2021	12:27P	DOUBLE PARKING	115	0	0	0	115	0	060	K	
4776	JPV5179	NY	PAS	8891052176	01/05/2021	11:26A	DOUBLE PARKING	115	0	0	0	115	0	073	K	
4899	87660KA	NY	COM	8823152124	02/22/2020	09:08A	DOUBLE PARKING	115	0	0	115	0	0	024	NY	
4902	JHZ4440	NY	PAS	8891057563	01/11/2021	11:54A	DOUBLE PARKING	115	0	0	0	115	0	081	K	

120 rows × 19 columns

*Table 10 Double parking Violations*

We can display the map using the following code

```
df = geopandas.read_file(geopandas.datasets.get_path('nybb'))
ax = df.plot(figsize=(10, 10), alpha=0.5, edgecolor='k')
```

This code plots the NYC map, and we can customize the points based on our required locations. All the counties in NYC can also be visible through this map.

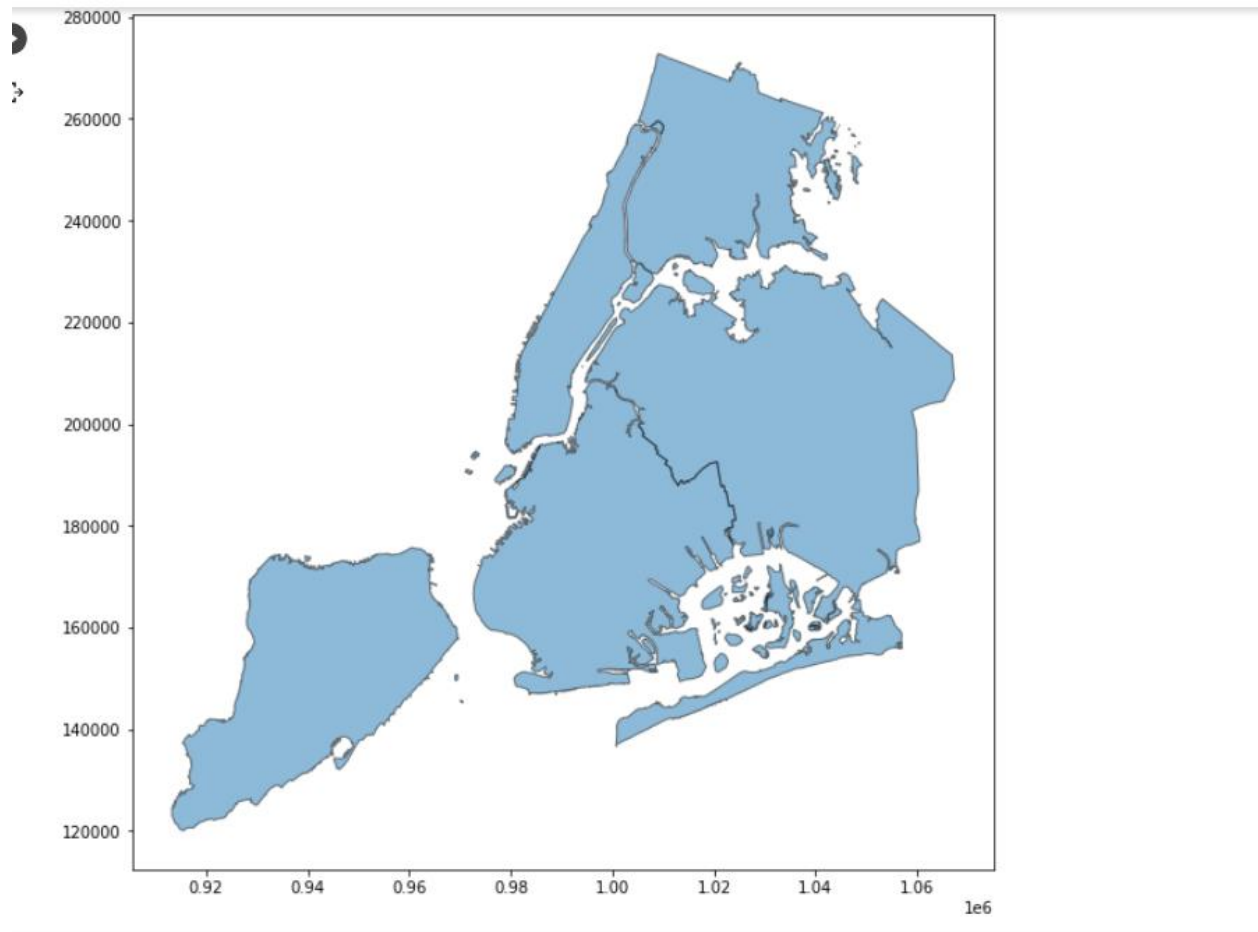


Figure 8 NYC Map

We can use this map to give values to the coordinates and to plot the points on the map. Also, we can attach the locations using some python libraries like geoplots or geopy. By this we can also find out where most of the violations are taking place.

We can also find the Percentage of each type of Violations by the following code

```
# Count the number of parking violations in each county.
results_df = results_df.groupby('violation')['county'].count().reset_index(name='Percentage')

# Calculate the number of parking violations in each county as a percentage of all parking violations.
results_df['Percentage'] = results_df['Percentage'] / results_df['Percentage'].sum() * 100
```



# Sort and display the resulting dataframe.

```
results_df.sort_values(by='Percentage', ascending=False).reset_index(drop=True)
```

The above code displays the result as

	violation	Percentage
0	PHTO SCHOOL ZN SPEED VIOLATION	35.912497
1	NO PARKING-STREET CLEANING	15.819323
2	FAIL TO DSPLY MUNI METER RECPT	10.532712
3	NO STANDING-DAY/TIME LIMITS	5.225846
4	INSP. STICKER-EXPIRED/MISSING	4.922017
5	FIRE HYDRANT	4.476403
6	NO PARKING-DAY/TIME LIMITS	3.504152
7	EXPIRED MUNI METER	3.281345
8	DOUBLE PARKING	2.430626
9	REG. STICKER-EXPIRED/MISSING	2.309095
10	NO STANDING-BUS STOP	2.228074
11	NO STANDING-EXC. TRUCK LOADING	1.681183
12	FRONT OR BACK PLATE MISSING	1.336844
13	FAIL TO DISP. MUNI METER RECPT	0.749443
14	NO STANDING-COMM METER ZONE	0.668422
15	NO STANDING-EXC. AUTH. VEHICLE	0.607656
16	CROSSWALK	0.567146
17	BIKE LANE	0.486125

Table 11 Violation Percentage

This table shows the percentage for each of the violations, PHTO SCHOOL ZN SPEED VIOLATION tops the table with 35%. This also says that there are more chances that the tickets will be of the first three type of violations, which covers up to 60% of the violations.

## STEP 5: Presentation of the Data

Data must be presented somewhere to visualize and get the outputs. For getting better visualization, we use various python libraries like NumPy, matplotlib, pyplot etc.

Considering the top 3 types of violations and plotting it on the graph and displaying the result.

```
import numpy as np
import matplotlib.pyplot as plt

# creating the dataset
data = {'PHTO ZONE':1242, 'NO PARKING-
STREET CLEANING':791, 'Double Parking ':230}
violation = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(violation, values, color='maroon',
        width = 0.4)

plt.xlabel("Types of Violations")
plt.ylabel("No. of violation tickets")
plt.title("Tickets in various types of Violations")
plt.show()
```



Figure 9 Graph showing tickets in types of violations

Here, we are considering the type of violations and their Percentage. As we all know PHTO SCHOOL ZN SPEED VIOLATION has a greater number of tickets. So, here I am trying to take the percentage of those tickets and trying to plot on the graph.

In the below graph, I am assigning the percentages for each of the violation type as we got values in the above table.

```
import numpy as np

import matplotlib.pyplot as plt

# creating the dataset

data = {"PHTO SCHOOL ZN SPEED VIOLATION": 35.912, "NO PARKING-
STREET CLEANING": 15.819, "FAIL TO DSPLY MUNI METER RECPT": 10.532, "NO ST
ANDING-DAY/TIME LIMITS": 5.225, "INSP. STICKER-
EXPIRED/MISSING": 4.922, "FIRE HYDRANT": 4.476, "NO PARKING-
DAY/TIME LIMITS": 3.504, "EXPIRED MUNI METER": 3.281, "DOUBLE PARKING": 2.
430}

violation = list(data.keys())

values = list(data.values())

fig = plt.figure(figsize=(10, 5))
```

```
# creating the bar plot

plt.barh(violation, values, color='green')

plt.xlabel("Type of Violations")

plt.ylabel("Percentage of Violations")

plt.title("Percentage of all violations")

plt.show()
```

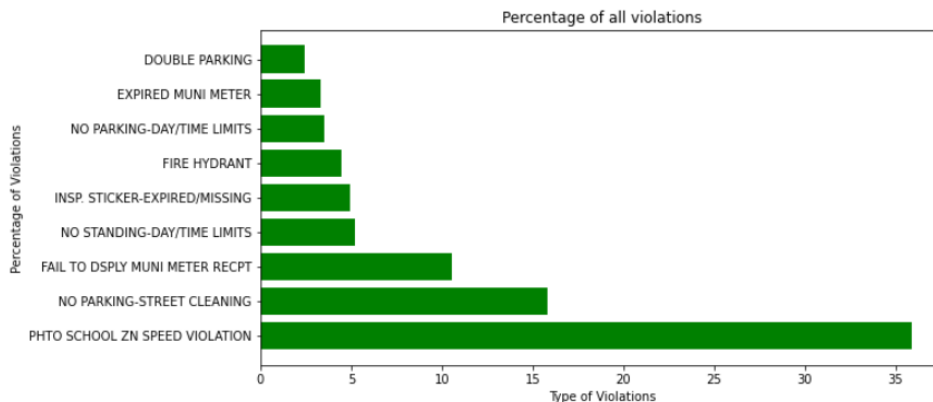


Figure 10 Graph showing percentage for types of violations

## CONCLUSION:

As a part of future research, I would like to produce all the ticket violations of the vehicles to be presented in the application similar like a user interface, which could display the area where more traffic violations usually occur. So, next time when some vehicle is passing by that way, they will be on alert.

This can also be done as a mobile application interface as well as in web interface.

# BIBLIOGRAPHY

- [1 "Google Colab notebook," [Online]. Available: <https://colab.research.google.com/>. [Accessed 14 12 2022].
- [2 "Open Data Law NYC," [Online]. Available: <https://opendata.cityofnewyork.us/open-data-law/#:~:text=On%20March%207%2C%202012%2C%20former%20Mayor%20Bloomberg%20signed,single%20web%20portal%20by%20the%20end%20of%202018..> [Accessed 17 12 2022].
- [3 Python, "Python," 2022. [Online]. Available: [www.python.org](http://www.python.org).]
- [4 "NYC Open data," 30 11 2022. [Online]. Available: <https://data.cityofnewyork.us/City-Government/Open-Parking-and-Camera-Violations/nc67-uf89>.
- [5 GitHub, "GitHub," [Online]. Available: <https://github.com>. [Accessed 19 12 2022].]
- [6 "Socrata API," [Online]. Available: <https://dev.socrata.com/foundry/data.cityofnewyork.us/nc67-uf89>. [Accessed 14 12 2022].
- [7 Geopandas, "Geopandas," [Online]. Available: <https://geopandas.org>. [Accessed 20 12 2022].]
- [8 PyPi, "PyPi," [Online]. Available: <https://pypi.org>. [Accessed 20 12 2022].]
- [9 Matplotlib, "Matplotlib," [Online]. Available: <https://matplotlib.org>. [Accessed 20 12 2022].]
- [1 Mapclassify, "Mapclassify," [Online]. Available: <https://pysal.org>. [Accessed 20 12 2022].0]
- [1 Pandas, "Pandas," [Online]. Available: <https://pandas.pydata.org>. [Accessed 20 12 2022].1]
- [1 NumPy, "NumPy," [Online]. Available: <https://numpy.org>. [Accessed 20 12 2022].2]
- [1 j. hawaii, p. [www.theopenbus.com](http://www.theopenbus.com), 30 11 2022.3]
- [1 "projects.betanyc.us," Beta NYC, NY, 2022.4]
- [1 "Socrata," 30 11 2022. [Online]. Available: <https://dev.socrata.com/> .5]
- [1 K. Okamoto, "Webology," pp. <http://www.webology.org/data-cms/articles/20200515050116pma142.pdf>, 30 11 2022.6]

- [1 K. Okamoto, "What is being done with open government data? An exploratory," 2016.  
7] [Online]. Available: <http://www.webology.org/data-cms/articles/20200515050116pma142.pdf>.
- [1 Q. Chen, "Parking for residential delivery in New York City: Regulations and behavior,"  
8] *ScienceDirect*, pp.  
[https://www.sciencedirect.com/science/article/abs/pii/S0967070X16302414?casa\\_token=OCCQAydUL5jAAAAAA:qHwCc9\\_cp5yuhJsL-LckzqtSVRF55TMALaj27onlyIQbXMD0wnXgGhH6gK\\_W5qxd3ID7MJM0w](https://www.sciencedirect.com/science/article/abs/pii/S0967070X16302414?casa_token=OCCQAydUL5jAAAAAA:qHwCc9_cp5yuhJsL-LckzqtSVRF55TMALaj27onlyIQbXMD0wnXgGhH6gK_W5qxd3ID7MJM0w), 2017.
- [1 W. Kim, "Double parking in New York city: a comparison between commercial vehicles and  
9] passenger vehicles," 2022. [Online]. Available:  
<https://link.springer.com/article/10.1007/s11116-021-10212-5>.
- [2 QuanquanChen, "Parking for residential delivery in New York City: Regulations and  
0] behavior," February 2017. [Online]. Available:  
[https://www.sciencedirect.com/science/article/abs/pii/S0967070X16302414?casa\\_token=EvJ8uJ5lsBUAAAAA:uRMtK50sO\\_wtu4WIXi1a1uX-dwxb\\_0XAhN\\_XUB22lcqpt3dRmYSDUQYEqqy572zgKyESte7IAbA](https://www.sciencedirect.com/science/article/abs/pii/S0967070X16302414?casa_token=EvJ8uJ5lsBUAAAAA:uRMtK50sO_wtu4WIXi1a1uX-dwxb_0XAhN_XUB22lcqpt3dRmYSDUQYEqqy572zgKyESte7IAbA).
- [2 webology, "webology.org," pp. <http://www.webology.org/data-cms/articles/20200515050116pma142>, 2022.
- [2 B. wilmingtion, "I Quant NY," *I Quant Tumbler*, p. [iquantny.tumblr.com](http://iquantny.tumblr.com), 2022.  
2]
- [2 S. Direct, p. <https://www.sciencedirect.com/science/article/pii/S0965856416310588>, 10  
3] 2018.

## GITHUB LINK:

<https://github.com/Pritesh-Yadav/MajorProject.git>