Data Intensive Computing Project Report - Phase 1

**Eco-Friendly but is it safe? Analysis and Predicting Trends in E-vehicle Accidents**

**Group members:**

Abhijeet Sanjiv Bonde (50624352)

Raghav Ratnakar Potdar (50631527)

Hui Yu (50311663)

**Problem Statement**

With the increasing adoption of E-scooters/E-bikes as a convenient and eco-friendly mode of transportation in New York City, concerns regarding their safety have emerged. This project aims to analyze and predict accident trends using historical motor vehicle collision data. Specifically, we will investigate the severity of E-scooter/E-bikes accidents compared to other vehicle types, identify key contributing factors, and determine high-risk locations and time periods for accidents.

**Data Sources:**

The data used for this project is the NYC Motor Vehicle Collisions (Crashes) dataset, which is an open-source dataset on the NYC Open Data Portal. The New York City Police Department (NYPD) keeps this data record which has extensive records of motor vehicle crashes that have been reported in New York City. The timeframe for the dataset is from June 2012 to Feb 2025 (Updated on a regular basis).

**Dataset Overview**

* Source: NYC Open Data (NYPD)
* URL: [NYC Motor Vehicle Collisions Dataset](https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/data)
* Number of Records: Over 2 million crash reports (updated regularly)
* Number of Columns: 29+ features that cover various aspects of each collision

**Data Attributes**

* The data includes information such as:
* Crash date and time
* Location details (borough, latitude/longitude, street names)
* Parties involved (vehicles, pedestrians, cyclists, motorists)
* Causing factors (distracted driving, speeding, failure to yield, etc.)
* Injuries and fatalities count
* Types of vehicles involved

**Data Quality**

* Missing Values: A lot of records have missing or incomplete fields (e.g., unspecified vehicle type or contributing factor with some misspelled entries).
* Human Error: Since data is manually recorded by officers or self-reported by drivers, there may be inconsistencies or biases.
* Time Lags: There may be a delay between when an accident occurs and when it is officially recorded in the dataset.

To curb these issues, the data needs to be cleaned and pre-processed before performing analysis.

**Data Cleaning**

**Kindly Do not run the DataCleaning2.ipynb and DataCleaning4.ipynb, since this is an API request module and would take 21 days to run on a single machine, they also generate Intermediate files which are attached inside the zip file.**

Initially, we started importing the required libraries for data cleaning such as pandas, numpy, and tabulate. We loaded the dataset into a DataFrame using the pandas.read\_csv() function. The size of the DataFrame is . We also check the datatypes of all the columns to have a better understanding of unformatted data, which will need extra steps to standardize the data format for each column.

1. **Filling missing data to null or 0.**
   1. Contributing Factors - Values in this column had some values which belong to Nan ("nan", "NaN", "None", "", " ", "N/A", "1", "80"). So, we replace them with unspecified value.
   2. Deaths and Injuries - Filled All NA values to 0 as if no death or injuries had occurred in the accident.
   3. Latitude and Longitude - Filled All NA values to 0. We require this step for further processing to fill in the latitude and longitude.

**Reason**: To ensure that no data is left empty while processing, we fill data with unspecified or 0. So it becomes easier to plot the graphs and missing values don’t hinder the analysis of data. Additionally, the missing values in Injuries and Deaths can make the graph skewed, therefore we assume that there were no casualties in the accident with missing Injury and Death values. Missing values in latitude and longitude can limit our geographical analysis, so we fill them with 0 (Further we try to fill in some of these missing values).

1. **Shifting Attributes**
   1. Car Type - The primary car for the accident is more important to analyze the data. VEHICLE TYPE CODE 1 has the primary car involved in the accident. So if data is missing for VEHICLE TYPE CODE 1 and if we have data in other TYPEs, we will swap the data according to it. This process is performed on all vehicle-type columns (1-5)
   2. E-Vehicle - Our problem statement mainly focuses on e-vehicle accidents. To better understand, we are shifting e-vehicles (e-bike, e-scooter) to VEHICLE TYPE CODE 1.

**Reason**: This step ensures that if there is a missing value in VEHICLE TYPE CODE 1 then it gets filled with any other VEHICLE TYPE CODE 2/3/4/5, which is required to reduce the false prediction in the future of vehicles as missing data could lead to incomplete analysis.

1. **Changing Case**
   1. Combining the "ON STREET NAME", "CROSS STREET NAME" and "OFF STREET NAME" to a new field called "Addresses - We are filling latitude, longitude, and zip code using nominatim API. We are required to combine these columns to make the address field. So, we are converting the data to lowercase and combining them to make a new column named Addresses. Also attaching the ‘new york’ at the end of the address which improves the API querying.
   2. Vehicle Types - Vehicle type code columns contain a mix of uppercase and lowercase. To remain consistent, we changed all values to lowercase.
   3. Contributing Factors - This column also had the data with different cases. So make it standardized, converting them into lowercase.

**Reason**: Changing the case of textual columns is necessary to avoid any duplicate categorization of values. Ex. Attention and attention are treated as different labels by the model. To avoid such conflicts and to maintain consistency among the data, we lower the case of all the textual columns.

1. **Fixing Date and Time** - The format of time and date in our dataset is inconsistent, which includes different symbols such as ‘\’ and ‘-’ to separate values. We used the panda’s built-in function 'to\_datetime' to standardize the date and time in our dataset.

**Reason**: To simplify the ‘time-based analysis’ and to match the standard format for date and time according to pandas to\_datetime() function we combine them and make a consistent column for crash date and time.

1. **Removing Duplicate Values** - Some entries have the same latitude, longitude, date, and vehicle car type with additional extra fields. So, here we are checking duplicate entries and sorting them

**Reason**: To avoid a skew in the graphs and its analysis we remove the duplicate values. This is also necessary to keep a single entry for each crash. Redundant data can result in biased outcomes which can lead to overfitting.

1. **Creating new columns**
   1. Create a new column based on crash time called ‘Time of Day’ – All the timestamps given in the dataset have 24-hour time, based on this we infer if the crash took place in the morning (5:00 to 12:00), afternoon (12:00 to 17:00), evening (17:00 to 21:00), or night (21:00 to 5:00). This helps to analyze the patterns in the crashes according to time of day.
   2. Create a new column called ‘is\_e\_vehicle’ in the DataFrame – If there is an e-vehicle involved in the crash then we set the value ‘True’, if there is no e-vehicle in the crash we set the value to ‘False’. This is done by checking ‘VEHICLE TYPE CODE 1/2/3/4/5’ columns and checking if there is an e-bike or e-scooter present.

**Reason**: This helps us analyze trends of crashes according to the time of day, throughout the day. This will further help us to predict the time of day when it is more likely for a crash to take place. Analyzing the is\_e\_vehicle column, it helps us distinguish between electric vehicles and other types of gas vehicles. Since the project focuses on e-vehicle accidents, it makes it easier to detect e-vehicle related crashes.

1. **Changing Data Types**
   1. Zip Code: The datatype of the zip code is float64, as zip codes are just region identifiers and their decimals don’t have any significance (14123.0 is the same as 14123), we convert them to Int64.
   2. Number of persons Injured – The number of persons Injured always has an integer value, it can either be that a person is injured or not, decimal values don’t make sense.
   3. Number of persons killed – The number of persons killed always has to be an integer, since either a person is killed or not killed, otherwise the person will be counted as injured, if alive. Hence decimal values don’t make sense.

**Reason**: As Int64 is easier to process and no information is lost, also some zip codes start with leading 0’s and storing them to float incorrectly identifies the region, we convert to int.

1. **Dropping columns**
   1. ON STREET NAME, CROSS STREET NAME, OFF STREET NAME – As discussed in step 3. We have created a new column called Address which has a combination of all the 3 attributes. We have already extracted useful information from their columns and combined it into the Addresses column, so dropping the column.

CRASH DATE, CRASH TIME – In the CRASH DATE & TIME column we have already filled in information in a standardized format. CRASH DATE and CRASH TIME have unstandardized information which we no longer require.

* 1. Dropping the rows that lie outside of New York - After fetching the data from API (in 3a.) There are some rows with values that lie outside of NY, since there are similar addresses that are in New York and outside of New York, So in order to preserve the data consistency we drop the columns. The dataset is focused on New York City vehicle collisions, so locations that lie outside of NY are miss-fetched locations and are irrelevant to the analysis and prediction. Including such data can introduce noise in the dataset which might lead to imperfect predictions. For instance, if we train the dataset to predict accidents based on location then including data that lie outside of New York would miss-train the model, forcing it to make erratic decisions, resulting in lower accuracy of the model.

**Reason**: Since we don’t want redundant information in our dataset, we remove the columns that already have a newly created derivative column which has extracted information from the deleted columns. This is helpful as this reduces the size of data which helps in faster data processing and also removes columns with unstandardized formatting.

1. **Filling missing values of Latitude and Longitude using external API**
   1. NLat, NLong, Location – We create 3 new columns called NLat, NLong, and Location then we utilize the previously created Addresses field to fetch the latitude, longitude, and a clean address from the ‘Nominatim’ API, which is an open street map. Since there is a limitation of 1 request per second, we deployed multiple instances from different IP addresses to fetch the data and fill the respective columns.
   2. BOROUGH – From the fetched data of Location (in 10a.) we extract the Borough information for New York from the fetched addresses and fill the BOROUGH by checking the possible values ('brooklyn', 'manhattan', 'queens', 'bronx', 'staten island')
   3. LATITUDE, LONGITUDE, LOCATION – From the fetched data (in 10a.) We check if there Is a location present in the LATITUDE and LONGITUDE columns, if there is a missing value, we extract the value from NLat and NLong columns which contain fetched data, and fill in the missing values.
   4. ZIP CODE – From the fetched data (in 10a.) we check if there is a Zip Code in the ZIP CODE column, if there is a missing value then we extract the zip code from the Location column which contains fetched addresses and fill it.
   5. ZIP CODE and BOROUGH (**Local Instance**) – Fetching more Zip code and Borough information from the local instance of ‘Nominatim’ as deployed on the docker instance and local instance have different output due to its filter differences. Leaving just 700 missing values approximately.

**Reason**: To maximize the geographical-based prediction, we try to fill in missing values from an open-source deployment. This will give more inputs for the model resulting in more accurate predictions and minimizing the biases. This will also help in understanding the most accident-prone geographical location.

1. **Standardizing vehicle types using domain knowledge** – Grouping similar and misspelled VEHICLE TYPE CODE’s into ‘suv’, ‘sedan’, ‘ambulance’, ‘passenger vehicle’, ‘truck’, ‘bus’, ‘bicycle’, ‘motorcycle’, ‘e-scooter’, ‘e-bike’, ‘law enforcement vehicle’, ‘van’, ‘taxi’, ‘moped’, ‘utility vehicle’, ‘construction vehicle’, ‘emergency vehicle’, and ‘unknown’ to remove inconsistency in the dataset. We ensure all the vehicle types are categorized in a uniform manner with no spelling mistakes, abbreviations or different naming types in the mapping.

**Reason**: This step is required in the Machine Learning models as structured and clean data increases the predictive accuracy. This is often termed as ‘Noise’ which can hinder the model's learning pattern. Removing the Noise, helps the model to learn about the pattern/relationship between VEHICLE TYPE CODE’s with other features. This process will help us gain better accuracy in predictions.

1. **Filling missing values using probability distribution** – We try to fill the ‘unknown’ value in VEHICLE TYPE CODE 1/2/3/4/5 using probability distribution. This is done by calculating the probability distribution by considering target vehicles (['sedan', 'sport utility vehicle', 'taxi']), then we calculate the normalized frequency distribution of the target vehicles which will calculate the distribution for all the vehicles present in the dataset. Then the number of rows that have that passenger vehicle is counted and if it is > 0 it replaces the value of unknown by the closest target vehicle type which is sampled randomly.

**Reason**: This step ensures that the missing data is filled with the most ‘appropriate’ data according to weight, which replaces a lot of missing values with a predicted value, which reduces the noise in the dataset as an ‘unknown’ vehicle type can reduce the accuracy of the model. Also, it makes the dataset more reliable, as this step also ensures that the data distribution is intact which does not decrease the stats of the dataset and also fills the accounted data. This step increases the yield of the dataset at the same time.

**Exploratory Data Analysis**

**Graph/Plot Overview:**

1. Heatmap - MV Accident Heatmap
2. Heatmap - EV Accident Heatmap
3. Scatterplot - MV Accident Scatterplot
4. Scatterplot - EV Accident Scatterplot
5. Scatterplot - All Fatality Locations
6. Scatterplot - MV Injury Locations
7. Scatterplot - EV Injury Locations
8. Bar graph - MV Injury vs Fatality
9. Pie chart - MV Fatality%
10. Bar graph - EV Injuries vs Fatalities
11. Pie chart - EV Fatality%
12. Bar graph - EV vs MV Injuries by Borough
13. Scatterplot - MV Injury Locations by ZIP Code
14. Scatterplot - EV Injury Locations by ZIP Code
15. Bar graph - MV Contributing Factors
16. Bar graph - MV Contributing Factors
17. Pie chart - MV Top Contributing Factors
18. Pie chart - EV Top Contributing Factors
19. Bar graph - MV Crashes Per Year
20. Bar graph - EV Crashes Per Year
21. Histogram - Identify Relationships Between Time And Crash Occurrence
22. Histogram - Identify Relationships Between Time and Crash Occurrence (EV)
23. Bar graph - Crashes by Day of Week and Time of Day (EV)
24. Histogram - Crash Occurrence and Severity by Time of Day (All Crashes)
25. Bar graph - Severity of Incidents For Top 10 Vehicle Type

**Note**: Electric Vehicles(EVs) in the context of this project meant low-speed electric vehicles, such as e-scooters and e-bikes, and Motor Vehicles(EVs) or Non-EV meant vehicles that do not fall into the category of low-speed electric vehicles.

**Possible bias**: The dataset we’re using only shows recorded accidents in New York City and unreported/unrecorded data are not considered in this project.

**1. Mapping Accidents Part 1**

This EDA step maps accident locations using folium heatmaps to identify accident patterns for all vehicles and EVs.

Graphs 1 and 2 show the intensity of the number of accidents on the map for motor vehicles and EVs respectively. Due to the massive amount of data we have, both heat maps are overcrowded providing little to no useful information when zoomed out, but since both heat maps are interactive, zooming in for both maps can help us find hot spots on the street level. For example, intersections and areas near highways are more likely to experience accidents. On the other hand, the second heat map for EVs shows an interesting fact about the borough of Staten Island where the frequency of accidents for EVs is relatively rare in Staten Island compared to other boroughs. We hypothesize that since Staten Island is almost entirely a residential area with long distances between areas of interest, using EVs to travel around is too inefficient. Compared to more crowded areas like Manhattan where EVs are more likely to be utilized as a result of shorter distances between areas of interest and narrower streets which discourage motor vehicle usage. Another possibility that explains the lack of EV accidents in Staten Island could simply be that Staten Island provides much safer driving conditions for EVs, but we believe it’s unlikely.

There is so much more interesting information in these two heatmaps waiting to be discovered but due to the lack of time for this assignment, we can only obtain limited information from these two heatmaps.

**2. Mapping Accidents Part 2**

This EDA step takes a different view of mapping accidents by using seaborn scatter plots to identify hotspots.

Graphs 3 and 4 also show the intensity of the number of accidents of motor vehicles and EVs respectively but plotted on scatter plots. Even though we’re plotting on scatter plots, we can still clearly see the shape of New York City without a map background. Compared to heat maps, scatter plots are better used to get an overview of the entire city while heat maps provide more detailed reports. From graph 3 for motor vehicles, we can clearly see almost all major artery roads in the entire city as wider roads experience more traffic, and more traffic contributes to a higher chance of vehicle accidents. Some interesting facts we observed are highly concentrated accidents around bridges and tunnels. We hypothesize that due to multiple roads and highways merging into these choke points, drivers are more likely to be confused due to merging, picking the right lane or exit, and the sheer amount of traffic going through these choke points.

Another interesting fact about traffic in Manhattan is that the Metropolitan Transportation Authorities(MTA) recently pushed out the Congestion Pricing Program on Jan 5, 2025, marking areas under the central park as a Congestion Relief Zone, charging tolls to discourage drivers from entering the area. Looking at our graph 3, we can clearly see the concentrated amount of traffic accidents in lower Manhattan below the central park showing how chaotic that area is. With this map, we hope to ease some tensions between citizens in NYC and the MTA for changing tolls for entering lower Manhattan by providing a visualization of why the MTA rolled out the program.

Lastly, graph 4 shows clusters of accidents around major population areas such as Astoria, Harlem, South Bronx, and Downtown Brooklyn, and the lack of accidents around highways because traffic rules prohibit EVs from getting onto highways. We hypothesize that EVs are more popular in these major population areas due to reasons described previously on the efficiency of EVs in crowded areas which contributes to the high likelihood of EV accidents. Interestingly, we hypothesized Lower Manhattan to also share concentrated EV accidents just like other population-dense areas but our scatter plot shows otherwise. This might be due to stricter rules and regulations in Lower Manhattan and/or the high density of traffic/pedestrians leaving no room for EVs.

**3. Fatality and Injury Distribution**

This EDA step focuses on analyzing the number of fatalities and injuries in accidents, comparing trends between EVs and motor vehicles.

Graph 5 shows the number of fatal accidents of all vehicles across NYC. We’re surprised by the low number of fatalities distributed across NYC considering the number of accidents that occurred and a number of injuries which are shown in Graph 6. Some interesting locations including the Bell Parkway above JFK airport and Interstate 278 in Brooklyn show concentrated fatalities among those highways. We can not explain why these areas are more likely to experience fatal accidents other than heavy traffic in these areas.

Graphs 6 and 7 show the number of injuries in motor vehicles and EVs respectively. Interestingly, we calculated the percentage of accidents of both motor vehicles and EVs that resulted in injuries. We found that about 23.9 percent of motor vehicle accidents result in injuries while about 78.4 percent of EV accidents result in injuries. We believe that advanced safety features in motor vehicles such as airbags, steel frames, and seatbelts play a key role in keeping the accident-to-injury ratio low. As for EVs, we believe that the lack of safety features or people’s recklessness in choosing not to wear safety features such as helmets or other protective gear results in a high injury ratio.

**4. Borough-Level Comparative Analysis**

This EDA step examines accident data at the borough level, identifying how different boroughs experience injuries and fatalities.

By studying graphs 8-12, we see that the number of fatal accidents is dramatically less than the number of injuries for both motor vehicles and EVs which further supports our previous observation on the fatality scatter plot. Interestingly, Brooklyn experiences the highest amount of injuries and fatalities for both motor vehicles and EVs. On the contrary, Staten Island experiences the least amount of injuries and fatalities. Looking at the pie chart, Staten Island has the highest percentage of fatalities of total accidents in their respective borough motor vehicles and Bronx for EVs. The pie chart for EVs might not be totally accurate due to its low fatality rate which we hypothesized to be related to the nature of EV’s slower top speed. Why does Brooklyn have a higher injury rate than other boroughs and how does Staten Island manage to keep injuries low? We do not have an answer with our current data set. In the future, we can dive deeper into these interesting results by using more variety of sources.

**5. ZIP Code-Level Comparative Analysis**

This EDA step examines accident frequency, injuries, and fatalities at the ZIP code level, identifying localized risk areas that may not be visible in borough-level analysis.

In graphs 13 and 14, we tried to explore deeper into the relationships between the number of injuries and geolocation within NYC. As we discovered during our borough-level analysis, Brooklyn experiences the highest amount of injuries, and Staten Island with the least amount of injuries for both motor vehicle and EV accidents. Surprisingly, we see that the number of injuries in zip codes in Staten Island for motor vehicles shows a similar trend with Manhattan and the Bronx. We suspect that Manhattan and Bronx have more population density than Staten Island making the comparison between raw injury numbers of these boroughs deceiving. As for EVs, the trend of the number of injuries in zip codes in Staten Island is minimal, again supporting our previous hypothesis on EVs being less popular in Staten Island.

Another interesting fact we observed from these two graphs is the zip code's highest amount of injuries. Overall, zip code 11207 has the highest amount of injuries for motor vehicles, and zip code 11220 has the highest amount of injuries for EVs. With this information, we can explore more in the future with additional information on why these areas experience higher amounts of injuries and areas we can improve on such as road design and safety awareness.

**6. Accident Cause Analysis**

This EDA step analyzes the most common contributing factors behind vehicle accidents, comparing EV vs. motor vehicle causes using bar charts and pie charts.

In graphs 15-18, we hope to explore contributing factors of all accidents to discover patterns and possible ways to reduce the number of accidents. Since our dataset does not map contributing factors to a particular vehicle but rather mapped to the entire accident, we can not differentiate which contributing factors belong to EVs or motor vehicles. As a result, we see large similarities between motor vehicles and EV graphs. Nonetheless, we can conclude that distraction, following too closely, and failure to yield the right-of-way are the most common contributing factors to vehicle accidents. Interestingly, the pie chart in graph 18 shows that “passing or lane usage improper” is unique to EVs. Since traffic rules and regulations often regulate EVs to cruise on the edge of the road, between the center of the road and the curve, we hypothesize that when motor vehicles are performing left or right-hand turns, drivers are likely to miss EV drivers when evaluating if the road condition is safe to turn due to reasons such as blind spots. In addition, we also hypothesize that the current road design is not EV-friendly, and improving the design of roads to consider EV safety can dramatically improve the accident rate between motor vehicles and EVs.

**7. Yearly Trend of Vehicle Accidents.**

This EDA step counts the number of accidents based on each year, giving us a better insight into changes throughout the years.

In graphs 19 and 20, we wish to examine the trend of vehicle accidents throughout the years. Looking at the yearly trend for motor vehicles in graph 19, 2014 seems to have unusually low accident counts due to unknown reasons, and data after 2019 seems to take a dramatic change in trending. We suspect the shift in 2019 is due to the start of the global COVID pandemic but not sure how it impacts the data set. Due to pandemics, traveling was banned for a period of time which might explain the dramatic drop. The method of recording might also be affected by the pandemic as the government was going through a difficult period of time. A lot more hypotheses can be made on what happened between 2019 and 2020, but without additional outside information, we’re unable to come to a conclusion.

As for the yearly trend of EV accidents in Graph 20, we can clearly see the rise in EV popularity. As the number of accidents increases, the number of people driving those EVs must also be increasing which is why we concluded that EVs are becoming more and more popular. Since our dataset cuts off in early 2025 and we suspect some data in 2024 might be missing as they’re still being processed, we hypothesize that the popularity of EVs will only grow in the near future with an increasing amount of EV-related accidents if the government decides to do nothing to improve the situation.

**8. Identify relationships between time and crash occurrences**

This EDA step uses a histogram to identify the number of crashes per hour for all vehicles and e-vehicles.

Graph 21 shows that more crashes occur from 14:00 to 18:00, which shows that most accidents occur after office hours. This trend continues for E-Vehicles (Graph 22), which have the most crashes from 17:00 to 18:00. This aligns with the typical office closing time in New York City, which is 5:00 PM. There are fewer crashes during late-night hours (e.g., 12 AM to 6 AM). In Graph 23, We can observe that there are more crashes on Friday afternoon and evening, which translates to a higher traffic volume from the end-of-week rush, more social and recreational outings, and possibly driver distraction or fatigue. The fact that commuters are heading home, people are heading out for the weekend, and possibly more impaired driving may all contribute to the rise in collisions. The weekend has fewer accidents, which translates to less congestion over the weekend since commuters are fewer over the weekend.

**9. Crash Severity Analysis to Death**

To do this analysis, we have created a severity score as death is more severe than injuries. The weighting factor 5 for fatalities is chosen to emphasize the severity of fatal crashes. Also, we have grouped them into 4 categories.

Graph 24 indicates that even if there are less number of crashes at night they are the most severe crashes which shows rash driving is more prominent at night than during the day. We can see that mornings have fewer crashes than daytime hours. This may be because morning traffic, though congested, is more organized in its flow compared to the congestion that occurs during the evening rush. Driver fatigue and congestion can be significant factors for a higher number of crashes in the afternoon and evening.

**10. Severity of Incidents For Top 10 Vehicle Type**

We are using crash severity to gain more insights into car types. First, we have normalized the data, as there are more accidents involving sedans and SUVs. We can see that bus accidents are more fatal, as other vehicles which are involved can be more damaged. This is because they are big and heavy which may significantly affect pedestrians or other small vehicles and they are hard to handle on narrow roads which can lead to accidents.

Bicycle and E-Vehicle (e-bike and e-scooter) accidents are less fatal as their operating speed is less than motorcycles. Riders are more vulnerable through the absence of body protection, leading to injuries upon crashes than deaths.

Sedans, SUVs, taxis, trucks, and vans have a similar pattern which displays a predominantly minor severity pattern in crashes, with some moderate, severe, and critical cases.

**NYC Car Crash Dataset:**

Data: <https://catalog.data.gov/dataset/motor-vehicle-collisions-crashes>

Alternative Link: <https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/data>

**References:**

* + - 1. M. Agrawal, ‘Pythonic Data Cleaning With pandas and NumPy’ [<https://realpython.com/python-data-cleaning-numpy-pandas/>]
      2. Vultr, ‘Python Numpy array astype() - Convert Data Type’ [<https://docs.vultr.com/python/third-party/numpy/array/astype>]
      3. Nominatim API, [<https://nominatim.org/release-docs/latest/>]
      4. A. Sharma, ‘Guide to Geospatial Analysis using Folium’, [<https://www.analyticsvidhya.com/blog/2020/06/guide-geospatial-analysis-folium-python/>]
      5. Folium, [<https://python-visualization.github.io/folium/latest/>]
      6. Leodanis Pozo Ramos, ‘Python's collections: A Buffet of Specialized Data Types’, [<https://realpython.com/python-collections-module/>]
      7. Seaborn, [<https://seaborn.pydata.org/>]
      8. Matplotlib, [<https://www.w3schools.com/python/matplotlib_plotting.asp>]
      9. Aniruddha Bhandari, ‘A Beginner’s Guide to matplotlib for Data Visualization and Exploration in Python’, [<https://medium.com/analytics-vidhya/a-beginners-guide-to-matplotlib-for-data-visualization-and-exploration-in-python-3fb32d03c3cd>]