Traffic Collisions : Montgomery Dataset Analysis

Aasim Inamdar
Master of Science in Data Analytics,
School of Computing,
National College of Ireland
Dublin, Ireland
x23236108@student.ncirl.ie

Sana Shafiq Jalgaonkar
Master of Science in Data Analytics,
School of Computing,
National College of Ireland
Dublin, Ireland
x22237941@student.ncirl.ie

Waleed Bin Umer
Master of Science in Data Analytics,
School of Computing,
National College of Ireland
Dublin, Ireland
x23187956@student.ncirl.ie

***Abstract*—Traffic safety is becoming a major concern for all the communities throughout the world, including Montgomery County in Maryland state of the USA. To give a thorough picture of the traffic collisions, the study examines the effectiveness of considering and analyzing three main datasets of an Automated Crash reporting System – ACRS used by Montgomery County to track the details about day-to-day traffic incidents within the county. The study presents an in-depth analysis of traffic collision data of Montgomery County. This analysis uses an Extract, Transform and Load (ETL) pipeline, following which the study does Exploratory Data Analysis and visualizations to inform and conclude urban planners on to develop targeted strategies for minimizing the impact of traffic collisions, and thus enhancing public safety and transportation facilities.**

The Integrated analysis of the ACRS datasets utilizes MySQL as the source database, Python Pandas as the Staging area, and MongoDB as the destination. After the ETL pipeline, the study does the visualizations with Python's pandas' library and other visualization tools like matplotlib and seaborn. The study aims to find some insights into traffic collisions into aspects such as weather conditions, drivers characteristics and road conditions.

Keywords—Traffic Collisions, Montgomery County, Python, ETL, Visualizations, API, Automation, ACRS, MongoDB, MySQL

I. INTRODUCTION

Traffic incidents are the major public health and safety concern around the world, resulting in injuries, deaths and financial damages. With developments in vehicle safety and traffic control systems, incidents continue to happen at an disturbing rate. Solving this problem needs an innovative strategy based on thorough data analysis and rational decision-making.

The Montgomery County's datasets which are obtained from the Automated Crash Reporting System (ACRS) provided by the county's government offers to be a valuable resource for studying the incidents in-depth. The datasets provides plenty of information on incidents, those involved as well as the contributing factors. With this datasets, studies can obtain the understanding of the occurrence of the collisions and foster specific measures to improve road safety.

To help the study, an ETL meaning Extract, Transform and Load pipeline that uses a relational database- MySQL as the source database followed by Pandas as the staging area for the transformations and MongoDB as the destination database. The overall flow of the pipeline provides a seamless integration of direct data obtained from APIs and direct files, including manipulation of the data ensuring data integrity and consistency throughout the process of analysis.

II. RELATED WORK

The previous study[1] focuses on a subsection of four lane-highway in Prince George, British Columbia, Canada studying the crash data between a specific period of time. The study used classic rate analysis along with a neural network regression model taking factors like seasonal fluctuations, traffic behaviours, and impact of weather into consideration. Therefore the main aim of previous study was to predict the monthly incident frequency on the basis of the volume of rainfall, snowfall, and temperature data. The outcome showed that the temperature and snowfall had a major effect on the traffic volumes and incidents. The study indicated the need of taking changes in the seasons and weather conditions into consideration when analysing the highway safety, it also provides a view on improvements for active highway safety management in the areas with severe weather conditions.

Based on the another study[2] it proposed an architecture for collisions avoidance system based on edge computing and the low latency communication networks. The architecture included 3 major components: vehicle, network infrastructure, and edge computers. The actual idea was to create a simulation algorithm showcasing how a vehicle uses the network infrastructure to send data packets to the edge computers that include position, speed, timestamp and vehicle id. The edge computers maintain a database of vehicles in it's covering area which it constantly keep on updating with the incoming new data packets. The study made a point that with advances in the edge computing and low latency communication networks like 5G, it is now possible to install effective collision avoidance system which will be able to take care of a large number of vehicles in real-time, hence improving road safety.

III. METHODOLGY

A. *Datasets Selection*

This study's datasets comes from the official government site of Montgomery County [3] which maintains the data into various format like JSON, CSV, APIs that are collected by the Automated Crash Reporting System (ACRS) within the county. These datasets have been selected as it tracks all the information on the collisions in the County and provides the complete information required for the thorough analysis.

B. *Tech Stack*

- Python: is used as a primary language and a medium for handling an analysing data due to its simplicity and flexibility. The study uses many visualizations and processing libraries of python like NumPy, Pandas, Matplotlib, Seaborn. As for

linking with source an target database, the study uses pymongo to establish a connection with MongoDB and mysql-connector-python for getting the language linked to MySQL database.

- MySQL: A relational database link MySQL is the source of data flowing from API and exported data from the official government site of the county as it's well suited for the applications with complex queries.
- MongoDB: A NoSQL database like MongoDB is the target that is the destination database where the transformed and cleaned data is stored as it is free from any predefined schema requirement.
- Socrata API: an API provided by Socrata[5], enables developers to interact programmatically with datasets hosted on the platform by the detailed documentation, code samples, and developmental resources to assist users in getting started with the API.
- Luigi: a strong python utility used for creating an organization of data pipeline by defining workflows as tasks.

C. *Datasets Descriptions*

- Incidents Dataset: Provides general information about each incident as well as traffic information that occurred in Montgomery County, collected via Automated Crash Reporting System (ACRS). The dataset has been extracted from the API provided on the official government site of the county which is powered by Socrata [4]. The dataset identifier for incident dataset is: bhju-22kf. The total number of rows are 97458 which can be exported as a CSV file without API. As for API extraction via Socrata that rows limit is 1000. Each row is collision record.
- Drivers Dataset: Contains information about the drivers engaged in the traffic incidents on the county and local roadways in the Montgomery County as collected by the Automated Crash Reporting System (ACRS). The dataset has been extracted from the API provided on the official government site of the county which is powered by Socrata [4]. The dataset identifier for this dataset is: mmzv-x632. When exported the total number of rows are 172105 and 43 columns into a CSV file. As for API extraction via Socrata that rows limit is 1000. Each row is represented as a driver record.
- Non-Motorist Dataset: Includes information about the non-motorists like pedestrians and bicyclists engaged in the collisions on the county and local roads in the Montgomery County , tracked via Automated Crash Reporting System (ACRS). The dataset is directly exported from the official government site of the Montgomery County. The total number of rows are 5650 with 32 columns, where each row is represented as non-motorist.

- Common Features in all three datasets:

Feature	Description
Report Number	ACRS Report Number assigned to the incident.
Local Case Number	Case number from the local investigating agency for the incident.
Agency Name	Name of the investigating agency.
ACRS Report Type	Identifies crash as property, injury, or fatal.
Crash Date/Time	Date and Time of crash.
Route Type	Type of roadway at crash location.
Road Name	Name of road.
Related Non-Motorist	Type(s) of non-motorist involved.
Collision Type	Type of collision.
Weather	Weather at collision location.
Surface Condition	Condition of roadway surface.
Light	Lighting conditions.
Traffic Control	Signage or traffic control devices.
Driver Substance Abuse	Substance abuse detected for all drivers involved.
Non-Motorist Substance Abuse	Substance abuse detected for all non-motorists involved.
Latitude, Longitude, Loaction	Y, X coordinate of crash location, location
Cross Street Type	Roadway type for nearest cross-street.
Cross Street Name	Name of nearest cross-street.
Off-Road Description	Description of location for off-road collisions.
Municipality	Jurisdiction for crash location.

Figure 1. Features available in all 3 datasets.

- Features in Incidents datasets:

Feature	Description
Hit/Run	Unit - Vehicle left the scene resulting in a hit and run event.
Mile Point	Location - Mile point.
Mile Point Direction	Location - Mile point direction.
Lane Direction	Road/Area - Lane direction of travel.
Lane Number	Road/Area - Lane number of where the event occurred on
Lane Type	Road/Area - Type of roadway/area lane.
Number of Lanes	Road/Area - Number of lanes.
Direction	Location - Direction from mile point.
Distance	Location - Distance from mile point
Distance Unit	Location - Unit of measurement for mile point distance.
Road Grade	Road/Area - Roadway grade.
NonTraffic	Location - Recorded as a Non-Traffic event.
First Harmful Event	The first event of the collision.
Second Harmful Event	The second event of the collision (if applicable).
Fixed Object Struck	The fixed object struck by vehicle (if applicable).
Junction	The type of junction where the collision occurred.
Intersection Type	If the collision was intersection related, this field describes the intersection characteristics.
Intersection Area	Road/Area - Describes the interchange type, such as it being a thru roadway, ramp, or other related area types.
Road Alignment	The road alignment where the collision occurred.
Road Condition	The condition of the road when the collision occurred.

Figure 2 Features of incidents dataset.

- Features in Drivers datasets:

Feature	Description
Person ID	Unique identifier for this non-motorist.
Driver At Fault	Whether this driver was at fault.
Injury Severity	Severity of injury to this driver.
Circumstance	Circumstance(s) specific to this driver.
Driver Distracted By	The reason the driver was distracted.
Drivers License State	The state the driver's license was issued.
Vehicle ID	The unique identifier for the driver's vehicle.
Vehicle Damage Extent	The severity of the vehicle damage.
Vehicle First Impact Location	Vehicle - Location of vehicle area where first impact occurred on.
Vehicle Second Impact Location	Vehicle - Location of vehicle area where second impact occurred on.
Vehicle Body Type	They body type of the vehicle.
Vehicle Movement	The movement of the vehicle at the time of the collision.
Vehicle Continuing Dir	Vehicle Circumstances - Continuation direction of vehicle after collisions
Vehicle Going Dir	Vehicle Circumstances - Movement of vehicle before collision.
Speed Limit	Vehicle Circumstances - Local area posted speed limit.
Driverless Vehicle	Vehicle Circumstances - If the vehicle was driverless or not.
Parked Vehicle	Vehicle - Defines if the vehicle was parked or not at the event.
Vehicle Year	Vehicle - The vehicle's year.
Vehicle Make	Vehicle - Make of the Vehicle
Vehicle Model	Vehicle - Model of the Vehicle

Figure 3. Features of drivers dataset

- Features in Non-Motorists datasets:

Feature	Description
Person ID	Unique identifier for this non-motorist.
Pedestrian Type	Type of non-motorist
Pedestrian Movement	Movement of non-motorist at time of collision.
Pedestrian Actions	Improper actions by non-motorist.
Pedestrian Location	Location of non-motorist.
Pedestrian Obeyed Traffic Signal	Non-motorist response to pedestrian signal.
Pedestrian Visibility	Visibility of non-motorist.
Safety Equipment	Non-Motorist - Type of safety equipment if any was used.

Figure 4. Features of non- motorists' dataset

D. Data Architecture

Figure 5. represents the actual lifecycle of the process including the ETL pipeline and visualizations.

- The data for Incidents and Drivers is extracted via Socrata API in the form of JSON which is converted using pandas into a CSV file. As for Non-Motorists, the data is exported into a direct CSV file.
- These 3 CSVs – the original datasets are integrated into MySQL which is the source database in the ETL pipeline.
- Using Python, the connection is established with MySQL to extract data from tables to data frames which acts as a staging area where all the transformation takes place.
- Once the transformations are completed, 3 intermediate JSON files for each of the data frame is created which are loaded as 3 collections into MongoDB which is the target in the established pipeline.
- Finally for getting insights and performing final analysis the cleaned and processed data from MongoDB is fetched into data frames whereby using visualization libraries insights are generated.

E. Preprocessing Methods

a) Setup & Configurations: MySQL server and workbench along with MongoDB Atlas cluster, database and network access had been setup granting rights to users to connect to project cluster. The changes and work done by each individual could be tracked by github history. Installation of python libraries is essential for smoother execution of the code.

b) Using API to generate a CSV file: The government of montgomery county has uploaded the data collected via Automated Crash Reporting Sysytem (ACRS) onto the Socrata Platform enabling users to use an API endpoint to fetch data. After signing up to the socrata platform and using library sodapy a code generate the data file via API in JSON format which by using pandas dataframe which is converted to a CSV files for *Incidents dataset and Drivers dataset* which acts as the original dataset. Before generating the CSV, the study discards columns which starts with pattern ':@' using RegEx and pandas.

c) Limitations of API: Due to APIs limitation the datasets to study the analysis of Incidents and Drivers were restricted to a subset of 1000 records. The limitation was imposed by the APIs default settings, which limits the number of rows available for analysis.

It is vital to acknowledge this limitation and understand the study's findings and conclusion are based on a subset of data that could be analyzed within these data constraints. Despite the limitation, the study seeks to provide useful information and contribute to a better understanding of the subject matter.

d) Feeding Data into MySQL: Once the connection is established with MySQL using python's MySQL-connector, a database named 'montgomery' is created inside which a table based on CSV's file structure is created and data of file is fed into table.

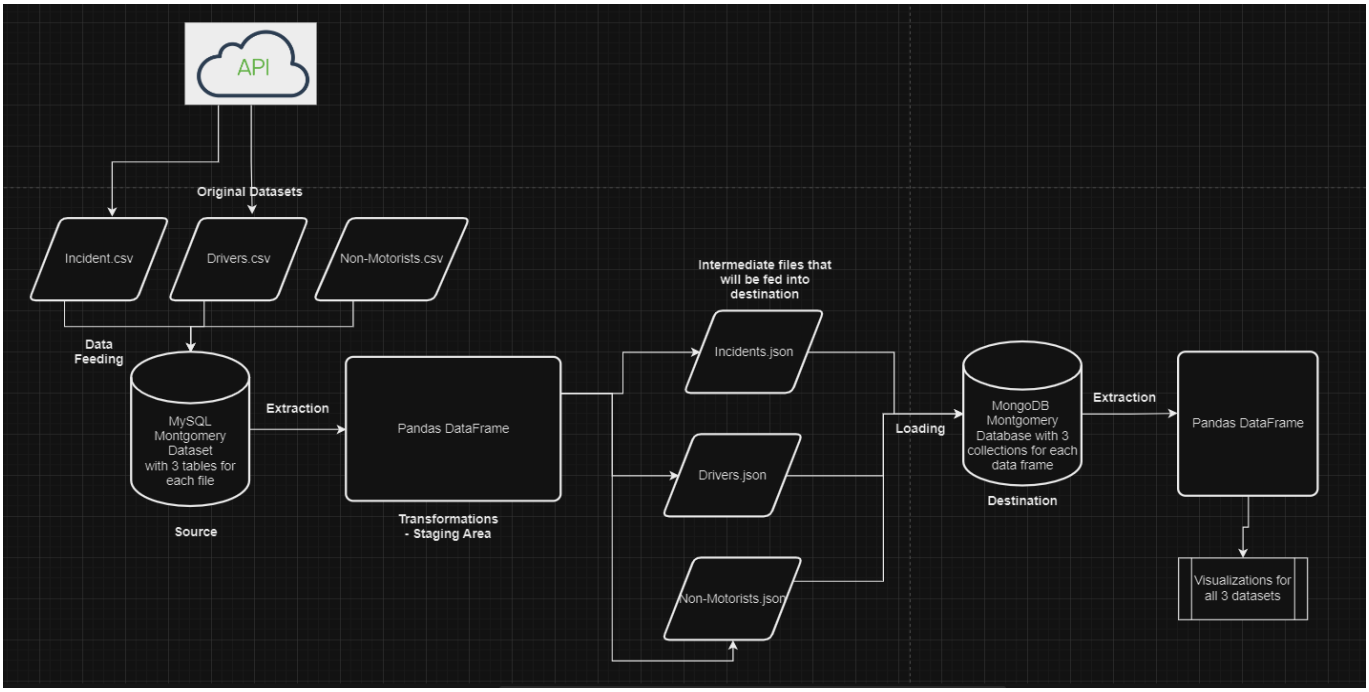

Figure 5. Data Architecture

e) Transforming Data in Pandas: The extraction is done using python where data from the table is extracted into pandas' data frame. Unnecessary Columns are dropped, and data is manipulated and finally stored into a finally cleaned data frame. This data frame is converted to a intermediate JSON file suitable for loading into MongoDB.

f) Loading the Intermediate file into MongoDB: A connection to MongoDB atlas cluster database is established using python. After creating a database and collection, the data from the intermediate JSON is loaded into the collections per file in the database.

g) Analysis and Visualization: The cleaned data loaded into the collection of the MongoDB is fetched back into pandas dataframe upon which analysis and visualization are performed with the help of Matplotlib and Seaborn to gain insigths about the collisions, drivers and non-motorists involved in the collisions.

h) Automation using Luigi: The study tried to implement the whole Extract Transform Load process has been automated with initializing extraction of data from API to generating a CSV file that needs to fed into MySQL to transformation at pandas level and final loading of cleaned data into the collections of MongoDB. The process generates a log file of the execution.

i) Leveraging jupyter notebook's 'restart kernel and run all' feature: Other way of automating the execution is to use the jupyter notebook's ' restart and run all' feature for the entire code per say every cell in the notebook to complete the execution without any troubleshooting and errors.

IV. Results and Evaluation

A. *Drivers:*

Looking at the data fetched from API based on drivers the following visualizations depict a clear picture of insights.

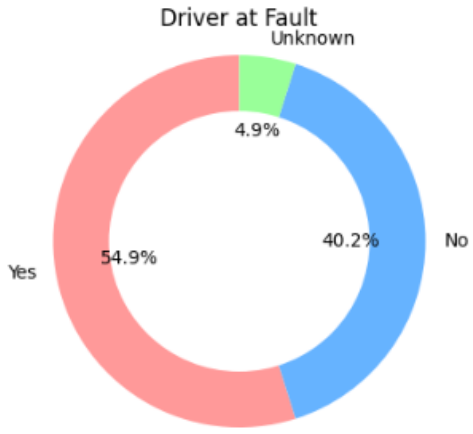

Figure 6. Drivers at fault.

1) Figure 6. Represents that almost 54.9% reports recorded the drivers fault in the collision .

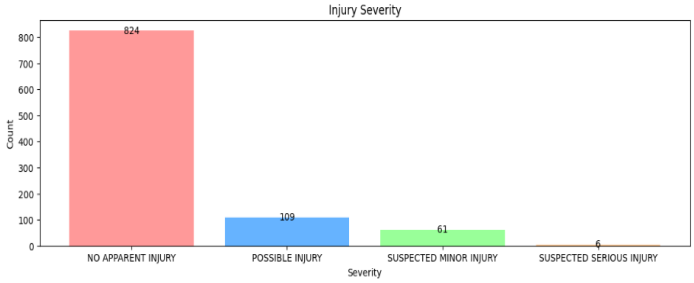

Figure 7. Injuries to Drivers.

2) Figure 7. Visualizes the driver's injury severity and its frequency of being the case in respect to the drivers involved in crashes. 824 out of 1000 cases had drivers apparently with no injury in the collisions whereas at least 109 drivers suffered any possible injury and 6 suspected to have suffered serious injuries in the collisions.

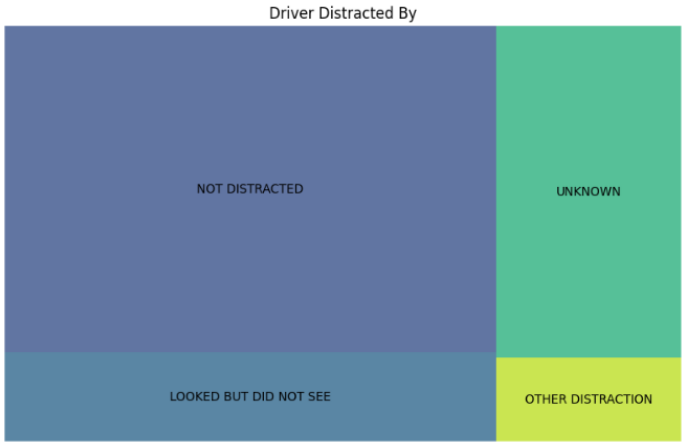

Figure 8. Cause accidents on drivers' part.

3) Figure 8. shows that in most of the cases the drivers were not distracted but ACRS also failed to record this parameter which is very important in determining the collision cause on the driver's part that might have led to an accident.

Figure 9. Word Cloud of Equipment problems.

4) Figure 9. depicts that equipment involved in the collisions were not misused. Also, the word cloud shows how in most of the cases for most records ACRS failed to track this parameter and alongside many unknown cases showcasing uncertainty regarding equipment problems.

B. Incidents:

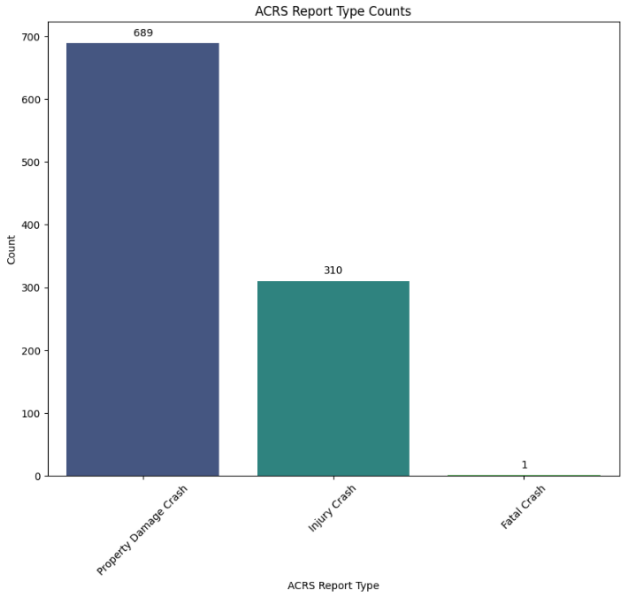

Figure 10 Type of ACRS Reports.

1) Figure 10 shows that most of the incidents resulted in property damage and a few in injury.

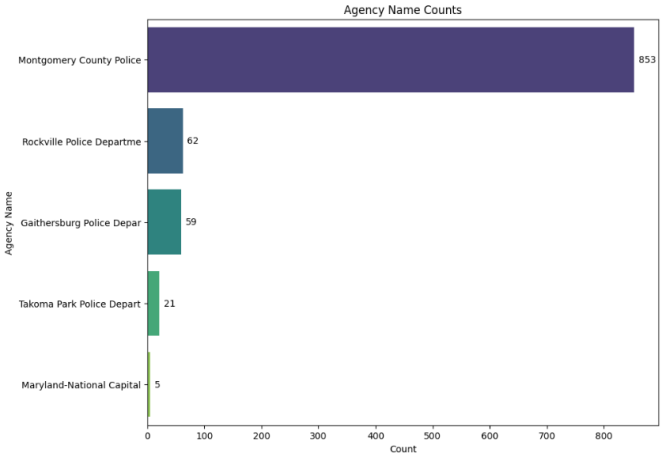

Figure 11. Records with Agencies.

2) The bar graph in figure 11. shows that Montgomery Couty's police agency has reported many cases as compared to other agency's followed by Rockville Police department and Gaithersburg Police Department.

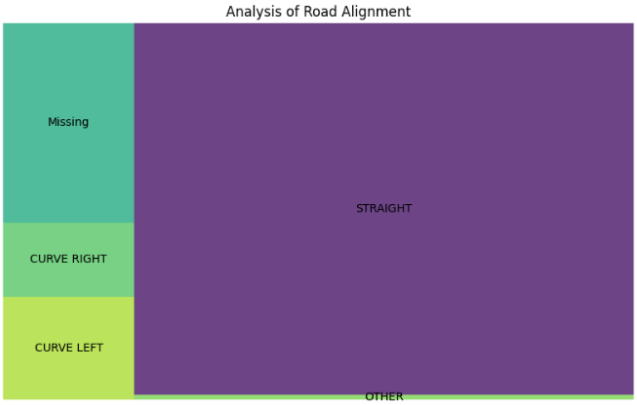

Figure 13. Road Alignment Analysis.

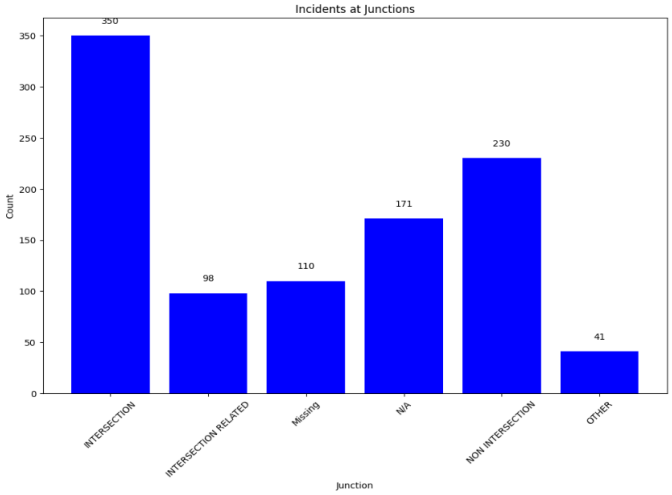

Figure 14. Rate of incidents at junction.

3) As seen in figure 13 and 15. Most of the collisions that have been recorded via ACRS were on the straight road with no defects. While this being the case, again it is important to note that we see ACRS failed to keep track of the parameters with respect to roads feature which has to be a cruicial factor for road planning and safety department in future planning.

4) It is clear from figure 14 that most of the incidents were at intersections which the road safety and planning department should make a note of in future planning. It is also found thet in most of the cases ACRS failed to keep the track the information as we see many values Missing and N/A which could lay the setback in future planning.

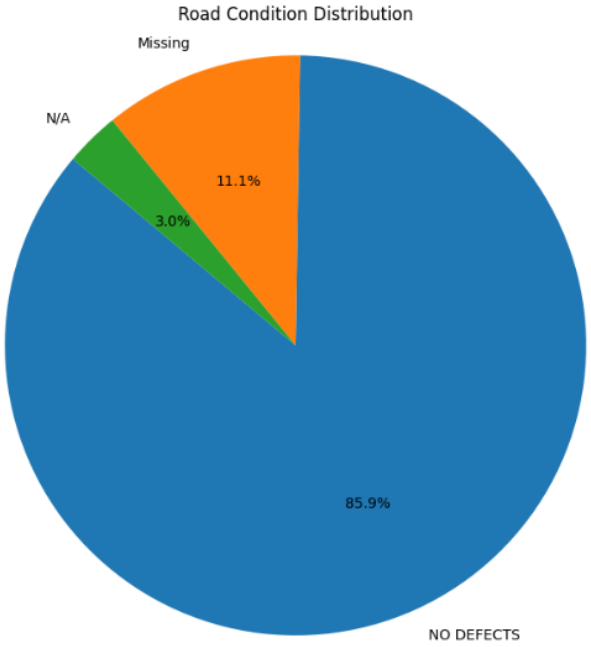

Figure 13. Effects of roads on incidents.

C. *Non-Motorists:*

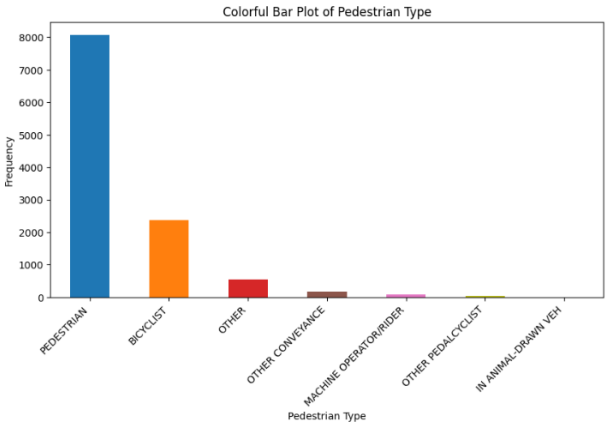

Figure 14 Types of Pedestrians.

1) It is clear from figure 14 that most of the non-motorists during a collision were pedestrians and bicyclists. A very few cases had been reported where any animal was drawn towards the vehicle that could have been led to a collision.

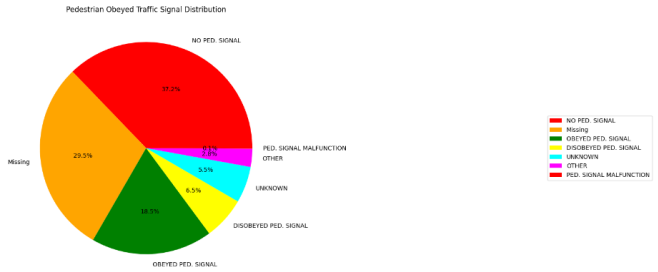

Figure 15. Traffic signal distribution

2) As seen in Figure 15. For most of the cases the pedestrians followed the signal but there were many more cases where ACRS failed to keep record of this important feature which is crucial to understand in any future matters for addressing the issues.

3) Figure 16. depicts how in many cases the at most of the time pedestrians were visible during the collision. It is clear that how most of the time pedestrians clothing is linked with their visibility at the time of incidents. Pedestrians with Mixed clothing and dark clothing weren't visible and resulted to be laest visible to drivers.

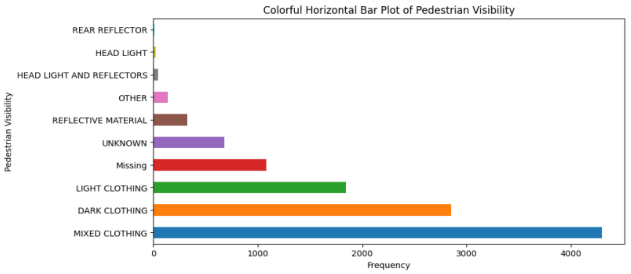

Figure 16. Pedestrian's visibility at time of incidents

4) Figure 17. Shows how most pedestrians happened to be crossing the roadway via crosswalk or at an intersection crosswalk at the times of collision. There should be some awareness or strict guidelines to avoid such things in the future.

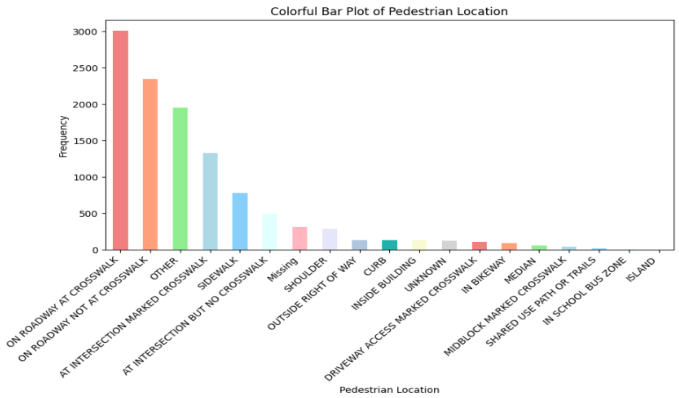

Figure 17. Pedestrian's location at time of collision

V. Conclusion and Future Work

In most of the cases it was drivers' fault, which was the cause of the incident, pointing out the importance of targeted measures to be taken into action like improving trainings of driving, enhancing traffic laws, and setting up the Advance Drivers Assistance System (ADAS) to reduce human errors on road.

Next, even though improper use of any equipment is not the common cause of the accident, the lack of detailed data on this feature points out how important it is to improve functionality of ACRS to track such features.

Effective road planning and safety precautions are challenged by the lack of data on road characteristics and driver behavior like distraction, despite high count of incidents at crosswalks. Future studies should be more focused on developing new procedures and techniques for complete data collection and revised analysis with the goal of making rational choices for traffic control and road planning design.

Future research should concentrate on using new technologies like Artificial Intelligence to improve the accuracy of handling such collisions data collection and analysis which will contribute to developing evidence-based methods for reducing the frequency of such collisions in future.

References

[1] Z. Luo, J. Li and M. Zhong, "Prediction of Seasonal Variation in Traffic Collisions on Urban Highway: A Case Study in the Province of British Columbia," 2021 6th International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 2021, pp. 104-109, doi: 10.1109/ICTIS54573.2021.9798674.

[2] R. Hasarinda, T. Tharuminda, K. Palitharathna and S. Edirisinghe, "Traffic Collision Avoidance with Vehicular Edge Computing," 2023 3rd International Conference on Advanced Research in Computing (ICARC), Belihuloya, Sri Lanka, 2023, pp. 316-321, doi: 10.1109/ICARC57651.2023.10145607.

[3] https://dev.socrata.com/

[4] https://dev.socrata.com/foundry/data.montgomerycountymd.gov/mmzv-x632 for Drivers API

[5] jbjbk

[6] hbhb

[7]