MACHINE LEARNING BASED ENHANCED ROAD SAFETY

PROJECT REPORT

CMPT 733 - BIG DATA LAB II

Team Members

Midhun Jisha Manoj

Risheek Sood

Shaik Sohail Hasan

Literature Review

In recent years, the application of machine learning techniques in urban transportation has emerged as a promising avenue for addressing critical challenges related to road safety, traffic rule enforcement, and parking management. This literature review delves into recent studies and advancements in leveraging machine learning to enhance safety and efficiency in urban transportation systems, with a specific focus on drowsiness detection, and license plate recognition.

❖ **Drowsiness Detection:**

Drowsy driving poses a significant risk to road safety worldwide. Recent studies have underscored the efficacy of machine learning algorithms in detecting driver drowsiness through the analysis of physiological signals and behavioral data. Face detection algorithm based on an extended set of Haar-like features is developed to automatically recognize the driver's face [1]. HOG is employed for accurate eye detection, complementing face detection for effective drowsiness detection [2]. These studies highlight the potential of deep learning models, such as convolutional neural networks (CNNs), in accurately identifying patterns of drowsy driving and providing timely warnings to drivers.

❖ **License Plate Recognition:**

Automated license plate recognition (ALPR) systems are instrumental in traffic rule enforcement by enabling efficient vehicle identification. Research demonstrates the utility of machine learning algorithms, By localizing number plates, detecting violations, employing Optical Character Recognition (OCR), and accurately extracting and recognizing license plate numbers from images captured by surveillance cameras [3]. Algorithm based on the YOLO model, incorporating a detection de-warping convolutional neural network (DU-CNN). The proposed Darknet-31 network enhances both feature extraction and extraction speed, with added network prediction scales to improve small license plate character detection [4]. These studies showcase the robustness of ML-based approaches in automating traffic enforcement processes, thereby enhancing monitoring capabilities and reducing human error.

❖ **Ticketing Analysis:**

There is an urgent need to enhance road safety, automate traffic rule enforcement, and improve parking management in urban areas. By harnessing machine learning algorithms for drowsiness detection and license plate recognition, alongside analyzing trends and patterns in tickets to uncover contributing factors and behaviors using data-driven insights, the project aims to significantly bolster road safety. Additionally, it aims to impact various other domains such as insurance and compliance checks, law enforcement, traffic management, Customs and Transportation Regulations, and Border Control, where these technologies can prove invaluable. By leveraging datasets comprising drowsiness metrics, parking ticket records, socio-economic data, driver demographics, and license plate images, the project aims to address key research questions pertaining to ticketing factors, the importance of license plate recognition for safety and automated enforcement, and the utilization of driver behavior data for proactive solutions. Through these endeavors, the project strives to create safer driving conditions and yield substantial benefits for drivers, passengers, pedestrians, businesses, government agencies, and personal users alike.

Problem Statement

In our quest to enhance road safety, we've pinpointed critical areas where improvements can make a substantial impact. These encompass:

Drowsiness Detection:

- ❖ Problem Description: Currently, there's a noticeable absence of automated tools that factor in driver characteristics. This raises the question: How can we leverage machine learning algorithms to detect driver drowsiness in real-time, thereby preventing accidents caused by drowsy driving?
- ❖ Challenge: Implementing real-time drowsiness detection using machine learning algorithms presents multifaceted challenges. Firstly, capturing and processing driver characteristics in real-time necessitates efficient data collection and processing mechanisms. Furthermore, accurately distinguishing between normal driving behavior and drowsiness patterns demands robust algorithm development.

License Plate Recognition:

- ❖ Problem Description: Accountability is key to ensuring safety, especially concerning vehicles. Thus, our focus is on identifying number plates to hold individuals accountable. How can we develop and implement a machine learning model to accurately recognize and extract license plate numbers from images for the automated enforcement of traffic rules and regulations?
- ❖ Challenge: Developing and implementing a machine learning model for accurate license plate recognition introduces numerous hurdles. Ensuring robustness to factors such as variations in plate sizes, fonts, and orientations requires extensive data preprocessing and feature engineering. Additionally, complications arise from handling occlusions, reflections, and poor image quality. Achieving real-time performance while maintaining high accuracy demands efficient model architectures and optimization techniques.

Ticketing Analysis:

- Problem Description: Understanding the significant factors contributing to tickets in Toronto is crucial. Moreover, analyzing these factors to uncover trends and patterns in parking ticket issuance can inform strategic deployment and optimization of human resources, reducing costs while enhancing performance.
- Challenge: Analyzing factors contributing to tickets and identifying trends and patterns entails navigating several challenges. Integrating and processing heterogeneous data sources such as parking ticket data, socio-economic data, and geographical information requires sophisticated data integration and cleaning techniques. Additionally, interpreting complex interactions between various factors influencing ticketing demands advanced statistical methodologies. Furthermore, deploying strategies based on analysis findings necessitates careful planning and coordination to optimize human resource deployment while minimizing costs.

These challenges, though formidable, present significant opportunities to revolutionize road safety and urban mobility through innovative applications of machine learning and data analysis techniques.

Data Science Pipeline

The overall data pipeline for our project looks like below:-

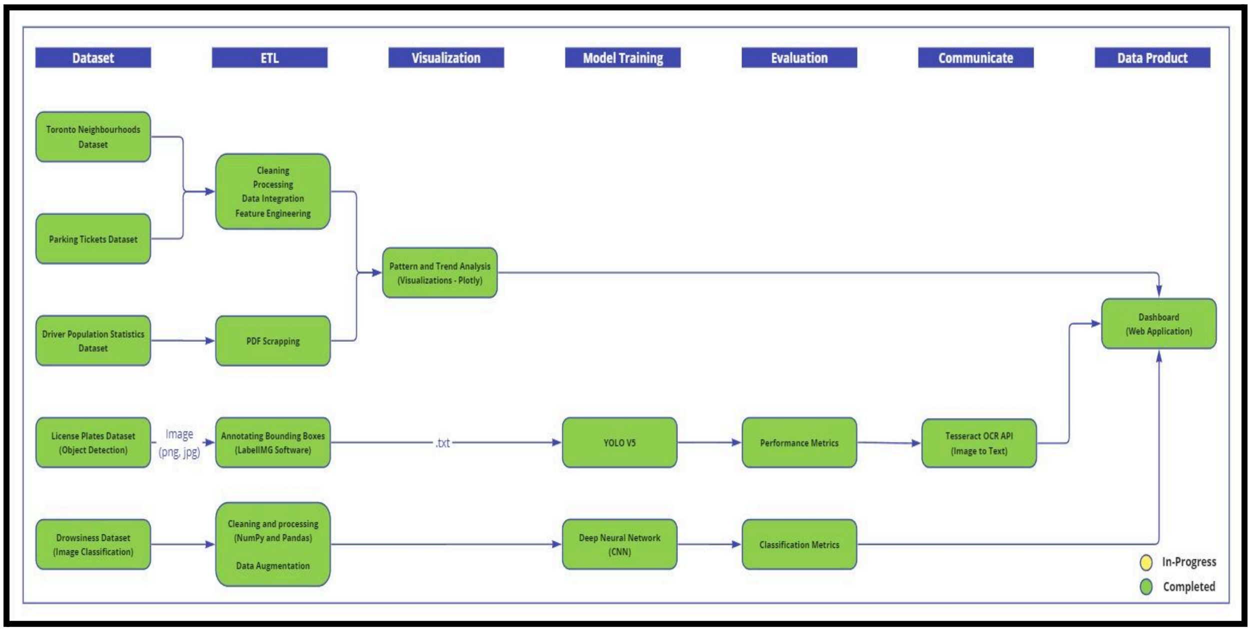

Dataset
ETL
Visualization
Model Training
Evaluation
Communicate
Data Product
Toronto Neighbourhoods Dataset
Parking Tickets Dataset
Cleaning Processing Data Integration Feature Engineering
Pattern and Trend Analysis (Visualizations - Plotly)
Driver Population Statistics Dataset
PDF Scrapping
Dashboard (Web Application)
License Plates Dataset (Object Detection)
Image (png, jpg)
Annotating Bounding Boxes (LabelIMG Software)
.txt
YOLO V5
Performance Metrics
Tesseract OCR API (Image to Text)
Drowsiness Dataset (Image Classification)
Cleaning and processing (NumPy and Pandas)
Data Augmentation
Deep Neural Network (CNN)
Classification Metrics
In-Progress
Completed

Fig 1: Pipeline

We started by selecting 5 datasets and below is a short description of the resource and the dataset :-

Drowsiness Dataset:

- Brief Description: There are images across 4 classes: closed eyes, open eyes, yawn, no yawn
- Source: https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset

Parking Tickets Dataset:

- Brief Description: Contains non-identifiable information relating to each parking ticket issued for each specific calendar year
- Source: https://open.toronto.ca/dataset/parking-tickets/

Toronto Neighborhoods Dataset:

- Brief Description: Socio-economic data for Toronto
- Source: https://open.toronto.ca/dataset/neighbourhoods/

Driver Population Statistics Dataset:

- Brief Description: Driver control and licensing annual statistics from the Ministry of Transportation of Ontario
- Source: https://data.ontario.ca/en/dataset/driver-population-statistics

License Plates Dataset:

- Brief Description: Images with bounding box annotations on car license plates
- Source: https://storage.googleapis.com/openimages/web/download_v7.html#download-manually

We initiated our project by harnessing the Toronto Neighbourhood's GeoJson file, a dataset outlining the boundary coordinates for Toronto neighborhoods. These coordinates, represented through latitude and longitude values, formed the backbone of our spatial data. Our next step involved integrating this geographical information with the Parking Tickets Dataset, a comprehensive compilation detailing issued tickets alongside their respective coordinates.

Our merging process was pivotal, hinging on the precise determination of whether the ticketing locations fell within the polygon coordinates delineating the boundaries of the neighborhood regions. Following this crucial merge, we embarked on a series of steps including cleaning, processing, integration, feature engineering, and visualization of the combined dataset.

Our overarching goal was to unearth intricate relationships and patterns embedded within various factors contributing to ticketing incidents. Through meticulous analysis, we aimed to unearth insights into the types, frequency, and spatial distribution of ticket issuance.

This comprehensive analysis served as the bedrock for our subsequent endeavors. Leveraging the Driver population statistics from the Ministry of Transportation of Ontario, we delved deeper into understanding driver demographics and licensing annual statistics. We scraped data with Tabula and then extracted vital insights from this data through meticulous processing and analysis.

Furthermore, we then tapped into the License plate dataset, which has images of vehicle's license plates, we utilized a software called labelImg where we manually annotated bounding boxes on the license plates and got the x_min, x_max, y_min, y_max, from this we calculated the center and total height and width. Then we used pre-trained YOLO model and further trained it on our supervised data which we created consisting of images along with a label file that contains details about the bounding boxes and class labels, which includes the class_id (0 for all license plates), center coordinates of the bounding box (center_x, center_y), the width, the height of the bounding box (all normalized by the image dimensions). This approach enabled us to achieve accurate license plate recognition.

The decision to opt for the YOLO model stemmed from its numerous advantages, including real-time processing capabilities, single-stage detection architecture, and versatility across various object detection tasks.

Moreover, we developed a driver drowsiness detection system using a Convolutional Neural Network (CNN) architecture trained on a dataset of images capturing various driver states like closed eyes, open eyes, yawns, and no yawns. The CNN model features Conv2D layers with decreasing filter sizes (256, 128, 64, 32) and MaxPooling layers after each convolutional layer for downsampling. To prevent overfitting, we included a Dropout layer with a dropout rate of 0.5. The final layers of the model consist of Dense layers for classification, with 64 neurons and a softmax activation function to classify into the four driver states. We trained the model using categorical cross-entropy loss and the Adam optimizer, monitoring accuracy as our primary performance metric. This CNN architecture has proven effective in accurately classifying driver states, making it an integral component of our comprehensive drowsiness detection system.

With these components in place, we embarked on the development of our web application. Combining HTML, CSS, Bootstrap, and JavaScript for the front-end and Python Flask for the backend, we crafted an intuitive interface showcasing our analysis results and machine learning models. Our web application boasted three tabs: Visualizations, License Plate Recognition, and Driver Drowsiness. Each tab offered unique functionalities, ranging from visualization of ticketing insights to real-time license plate recognition and driver drowsiness classification.

Our data product represents a culmination of rigorous analysis, cutting-edge machine learning models, and user-centric design. We firmly believe that our solution has the potential to revolutionize road safety through technologies such as license plate recognition and driver drowsiness detection. By providing early warning systems and automating enforcement, we anticipate a significant reduction in accidents and an overall improvement in safety for drivers, passengers, and pedestrians alike. Additionally, our analysis insights can inform strategic resource allocation, fostering enhanced traffic safety and efficiency.

Methodology

1) **GeoJson and Parking Tickets Dataset Integration:**
    - ❖ Tools and Libraries: Python, Pandas, Numpy, Shapely
    - ❖ Analysis Methods: Spatial analysis, merging datasets based on geographical coordinates.
    - ❖ Explanation: Python, along with libraries such as Pandas, NumPy, and Shapely, provides a robust toolkit for integrating GeoJson data and conducting spatial analysis. NumPy offers efficient numerical computations, which are essential for handling large datasets and performing mathematical operations. Shapely complements these capabilities by providing geometric operations and data structures, making it easier to work with geometric objects like points, lines, and polygons directly. Pandas, known for its data manipulation capabilities, becomes even more powerful when combined with NumPy and Shapely, allowing seamless integration of geospatial data and traditional tabular data. With Shapely, we can handle GeoJson files effortlessly and perform spatial analysis tasks such as checking if points fall within polygon boundaries. This capability was instrumental in merging the Parking Tickets Dataset with Toronto Neighbourhood's GeoJson file based on spatial relationships, enabling deeper insights and more comprehensive analysis.

2) **Data Cleaning, Processing, Integration, Feature Engineering, and Visualization:**

- ❖ Tools and Libraries: Python, Pandas, Matplotlib, Plotly
- ❖ Analysis Methods: Exploratory data analysis, feature engineering, data visualization
- ❖ Explanation: Python and Pandas were used for data cleaning, processing, and integration tasks. Matplotlib and Plotly are popular libraries in Python for creating visualizations. Exploratory data analysis (EDA) helped in understanding the dataset's characteristics, identifying patterns, and relationships among variables. Feature engineering involves creating new features from existing ones to improve model performance or gain insights.

3) **Driver Population Statistics Analysis:**

- ❖ Tools and Libraries: Python, Pandas, Tabula, Matplotlib
- ❖ Analysis Methods: Statistical analysis, demographic analysis
- ❖ Explanation: We used Tabula to scrape data from pdf. Further, Python and Pandas were used for data manipulation and statistical analysis. Demographic analysis aimed to understand driver demographics and licensing annual statistics to extract insights relevant to the project's objectives. Matplotlib is a popular Python library that we have used for creating visualizations.

4) **License Plate Recognition:**

- ❖ Tools and Libraries: Python, labelImg, CV2, Numpy, Pandas, Tensorflow, Pytesseract, scikit-learn
- ❖ Analysis Methods: Object detection (YOLO Model), deep learning
- ❖ Explanation: labelImg was used for annotating bounding boxes on license plates. YOLO is a deep learning-based object detection algorithm known for its speed and accuracy. By training the YOLO model on annotated license plate images, the model was able to accurately recognize license plates.

5) **Driver Drowsiness Detection:**

- ❖ Tools and Libraries: Python, Numpy, Pandas, CV2, Tensorflow, Matplotlib, scikit-learn
- ❖ Analysis Methods: Image classification (CNN Model), deep learning
- ❖ Explanation: CNNs are widely used for image classification tasks. By training a CNN model on a dataset comprising images of drivers with various states (e.g., closed eyes, open eyes, yawns), the model was able to classify images based on driver states accurately.

6) **Web Application Development:**

- Tools and Libraries: HTML, CSS, Bootstrap, JavaScript, Python, Flask
- Analysis Methods: Front-end and back-end development, user interface design
- Explanation: HTML, CSS, and JavaScript are used for building the front-end of the web application, providing the user interface and interactivity. Bootstrap is a front-end framework for designing responsive and mobile-first websites. Python Flask is a micro web framework used for building the back-end of the web application, handling requests and responses, and integrating with machine learning models and analysis results.

Overall, the chosen tools and analysis methods were selected based on their suitability for each specific task, considering factors such as efficiency, accuracy, and ease of integration within the project workflow.

Evaluation

For Ticketing Analysis: (For interacting with graphs please visit our website)

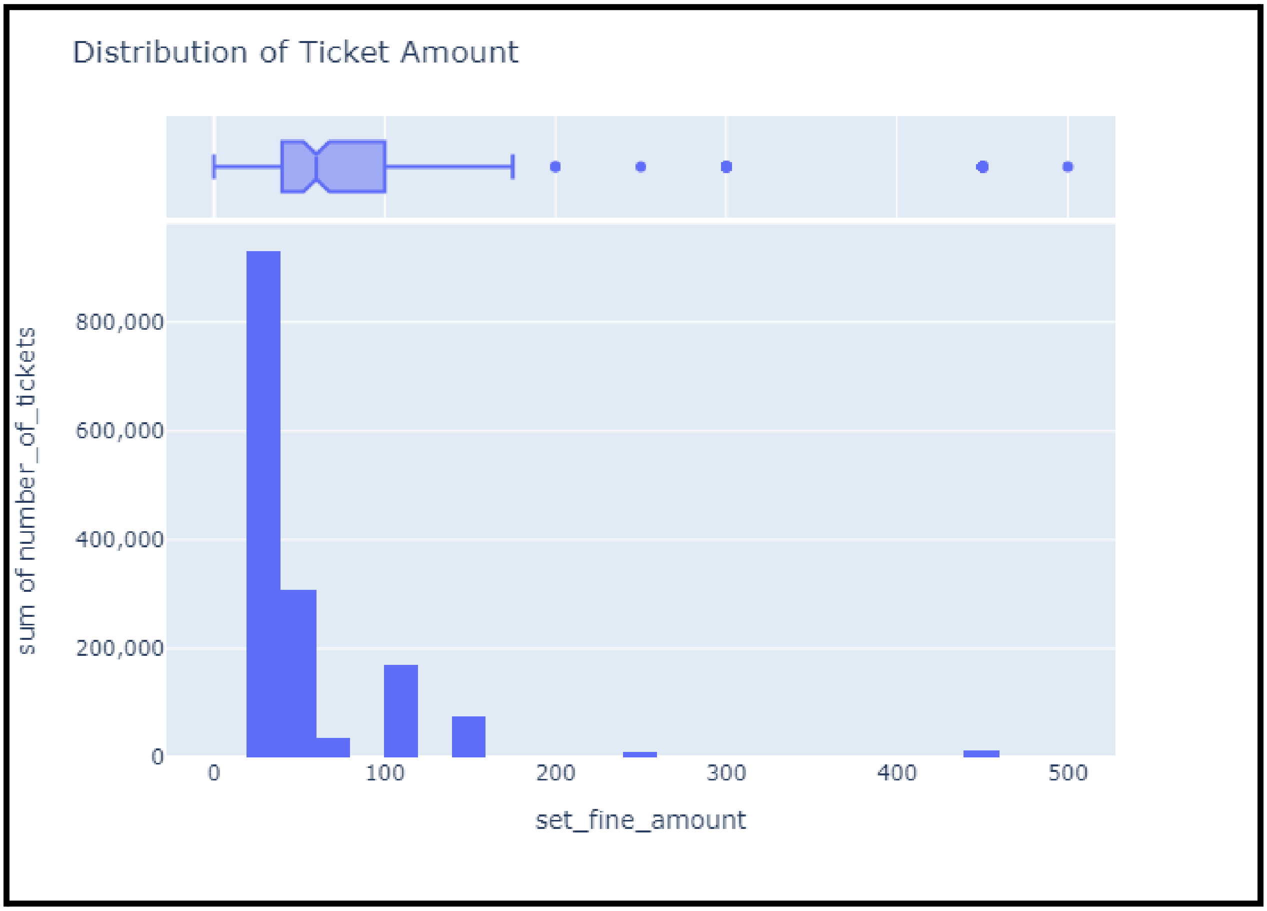


Fig 2: Distribution of Ticket Amount

This graph displays the distribution of ticket amounts, likely representing fines or penalties. The x-axis represents the set fine amount, while the y-axis shows the number of tickets issued. The most notable feature of the graph is the extremely high bar on the left, indicating that the vast majority of tickets were issued for the lowest fine amount, likely a standard minimum penalty. There are a few smaller bars further to the right, suggesting that some tickets were issued for higher fine amounts, though significantly fewer in number compared to the lowest amount. The graph clearly shows that most violations incurred the minimum penalty, with much fewer instances of higher fines being levied, potentially for more severe offenses or repeat violations.

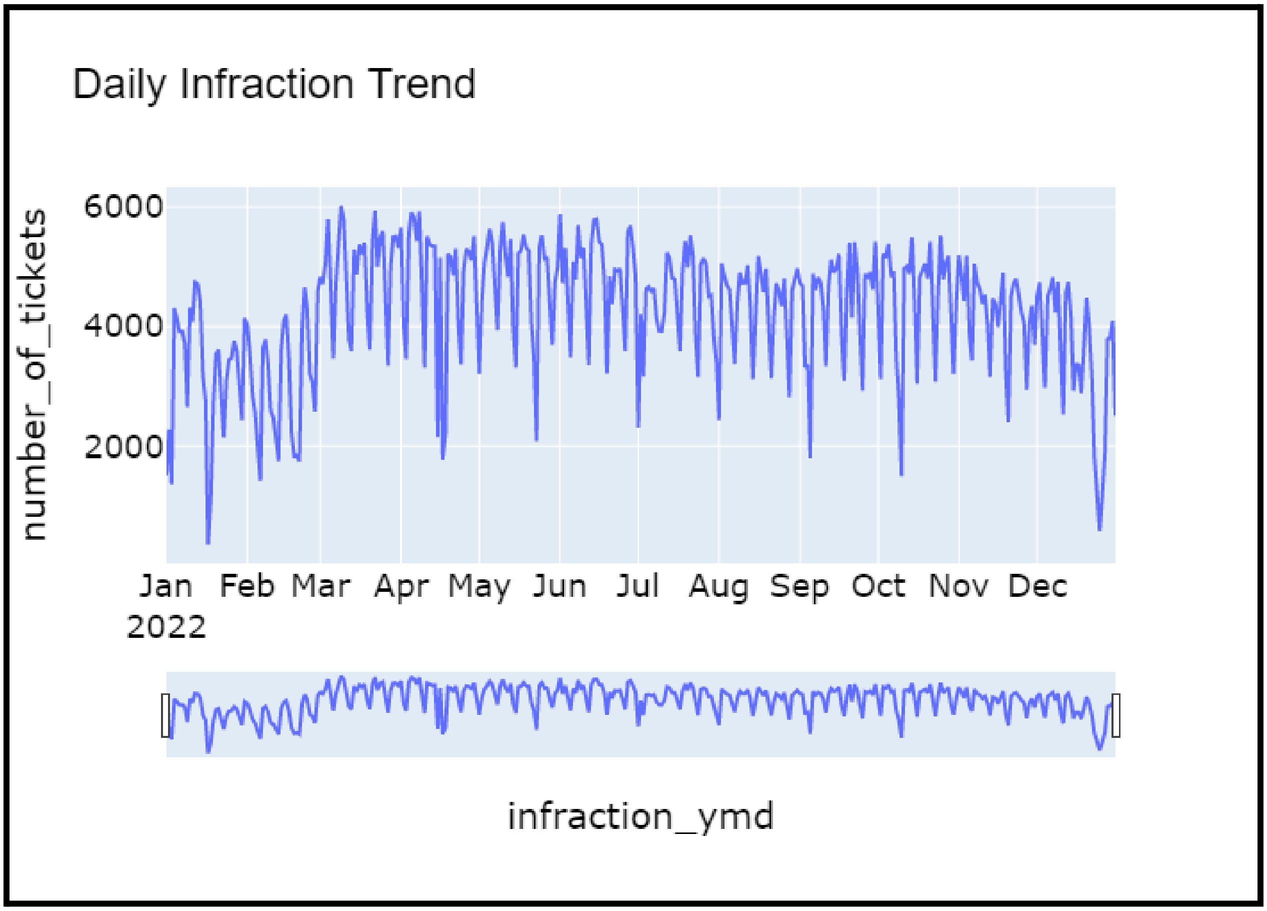


Fig 3: Daily Infraction Trend

The y-axis represents the number of tickets or infractions issued, while the x-axis displays the months of the year. There are periods of higher infraction activity, indicated by the taller peaks, which may correspond to certain seasons, holidays, or events that lead to increased violations. Conversely, there are also periods of lower infraction activity, represented by the troughs or dips in the graph, which could be due to factors like fewer people being out and about during certain times such as Mid Jan, April, Oct and so on. The overall shape and pattern of the graph suggest a recurring cycle, with infractions rising and falling in a somewhat predictable manner throughout the year. The magnitude of the fluctuations varies, with some peaks and troughs being more pronounced than others, indicating potential differences in the intensity or severity of infraction occurrences.

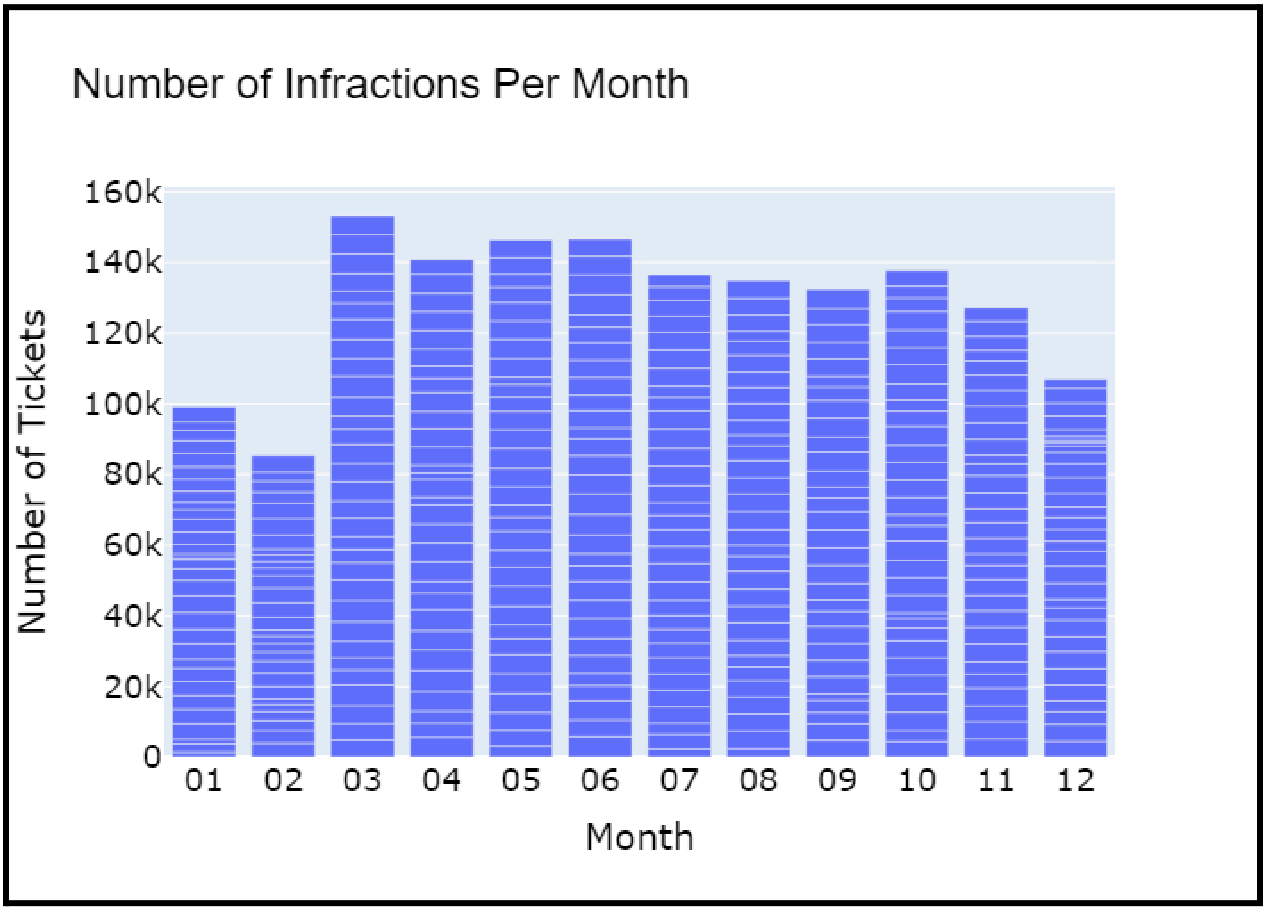


Fig 4: Number of Infractions per month

The above bar chart is displaying the "Number of Infractions Per Month." Each bar represents a month, labeled from 01 to 12, corresponding to January through December. The vertical axis represents the number of tickets issued, and we can see that the numbers on this axis range from 0 to above 150,000, increasing in increments. From the bar chart, it looks like the fewest tickets were issued in the 2nd month (February), while the most were issued in the 3rd month (March). The number of tickets tends to be high from March to October, with a noticeable decrease in the last two months, November and December. The pattern might suggest seasonal trends or changes in law enforcement or reporting practices throughout the year.

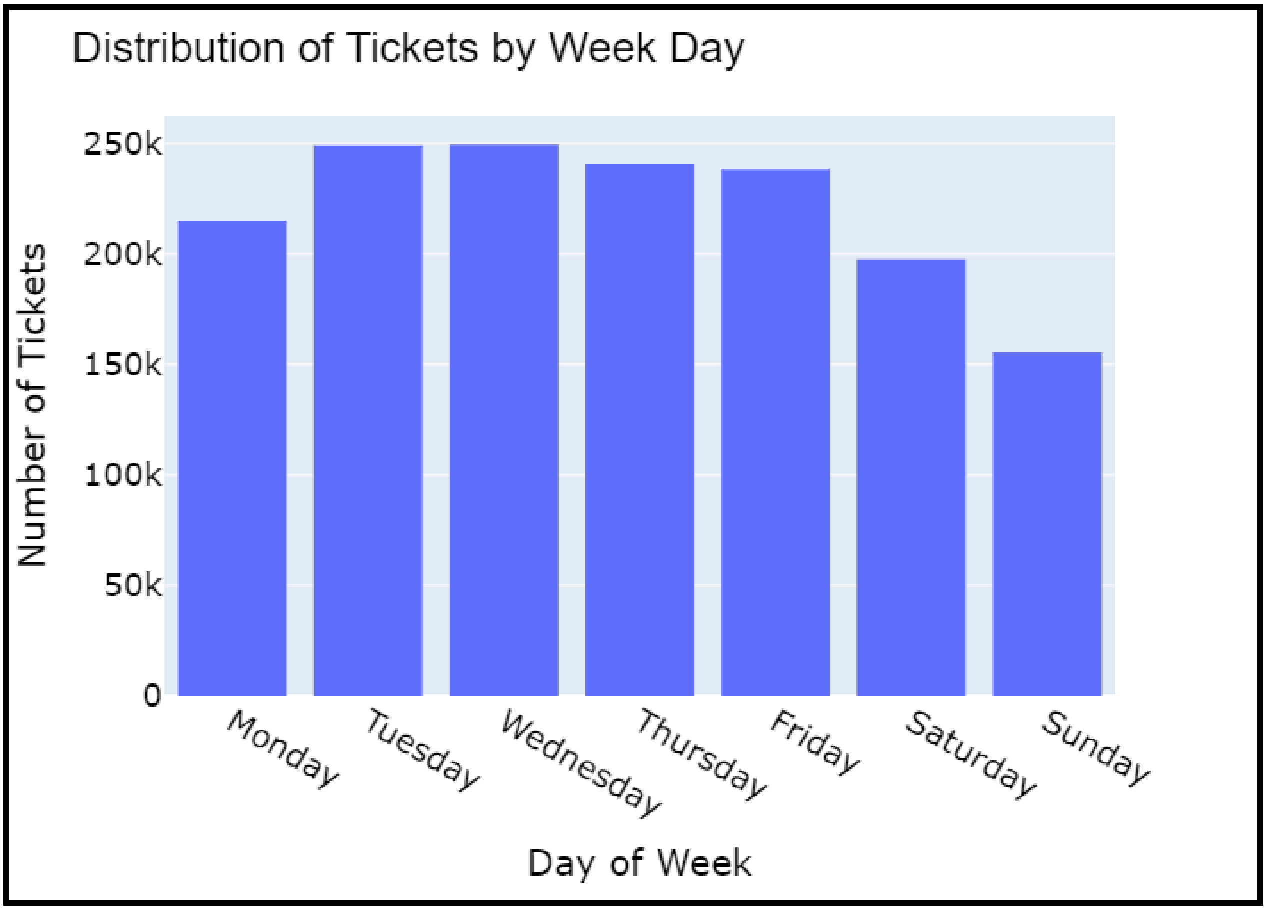


Fig 5: Distribution of Tickets by Week Day

X-Axis (Day of Week): Represents Monday through Sunday. Y-Axis (Number of Tickets): Ranges from 0 to 250,000. The daily infraction trend exhibits a clear pattern that corresponds to the days of the week. From Monday through Thursday, there is a consistently high volume of tickets issued, hovering around 250,000 each day. This suggests that during the typical workweek, when people are commuting and engaging in various activities, the incidence of infractions remains elevated and relatively steady. After Friday onwards, there is a slight dip in the number of tickets issued, close to the 250,000 mark. This could be attributed to a combination of factors, such as people leaving work earlier or engaging in different activities as the weekend approaches. Saturdays see a more noticeable decrease in infractions, with the number of tickets issued dropping below the Friday level but remaining above 200,000. This decline could be attributed to different activities on weekends, with fewer people commuting or engaging in activities. The most significant drop occurs on Sundays, with the number of tickets issued falling below 150,000. This substantial decrease likely reflects the reduced activity levels and different patterns of behavior that characterize Sundays for a significant portion of the population, such as fewer people on the roads or engaging in activities that could result in infractions.

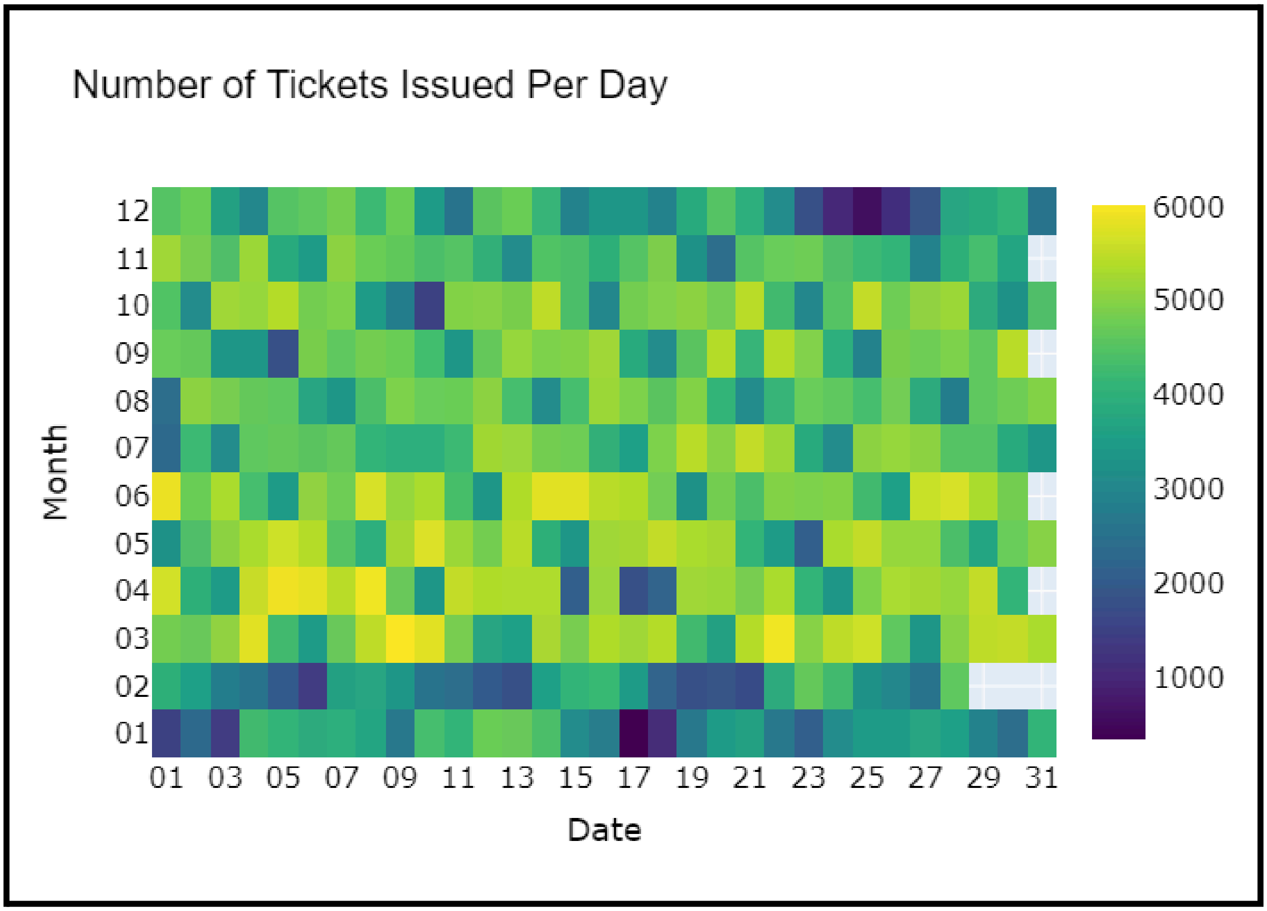


Fig 6: Number of Tickets Issued Per Day

This graph is a heat map that shows the "Number of Tickets Issued Per Day" across different months of a year. The horizontal axis represents days of the month, from 1 to 31, and the vertical axis represents months, from 1 to 12 (January to December). The colors represent the number of tickets issued, with the scale on the right side indicating the range of values. Cooler colors (like greens) represent fewer tickets, warmer colors (like yellows) represent more tickets, and the hottest colors (like purples) represent the highest numbers of tickets issued. Highest number of tickets are issued between March to June and Leaset are in Jan and February comparatively. Other months also near higher ranges of 4000 to 5000. With fewer days like 1, 5, 10, 17 to 21 being less, this can be due to various factors. One of the most common things we can see is in December during 23 to 26, where the tickets are less. This could be due to christmas.

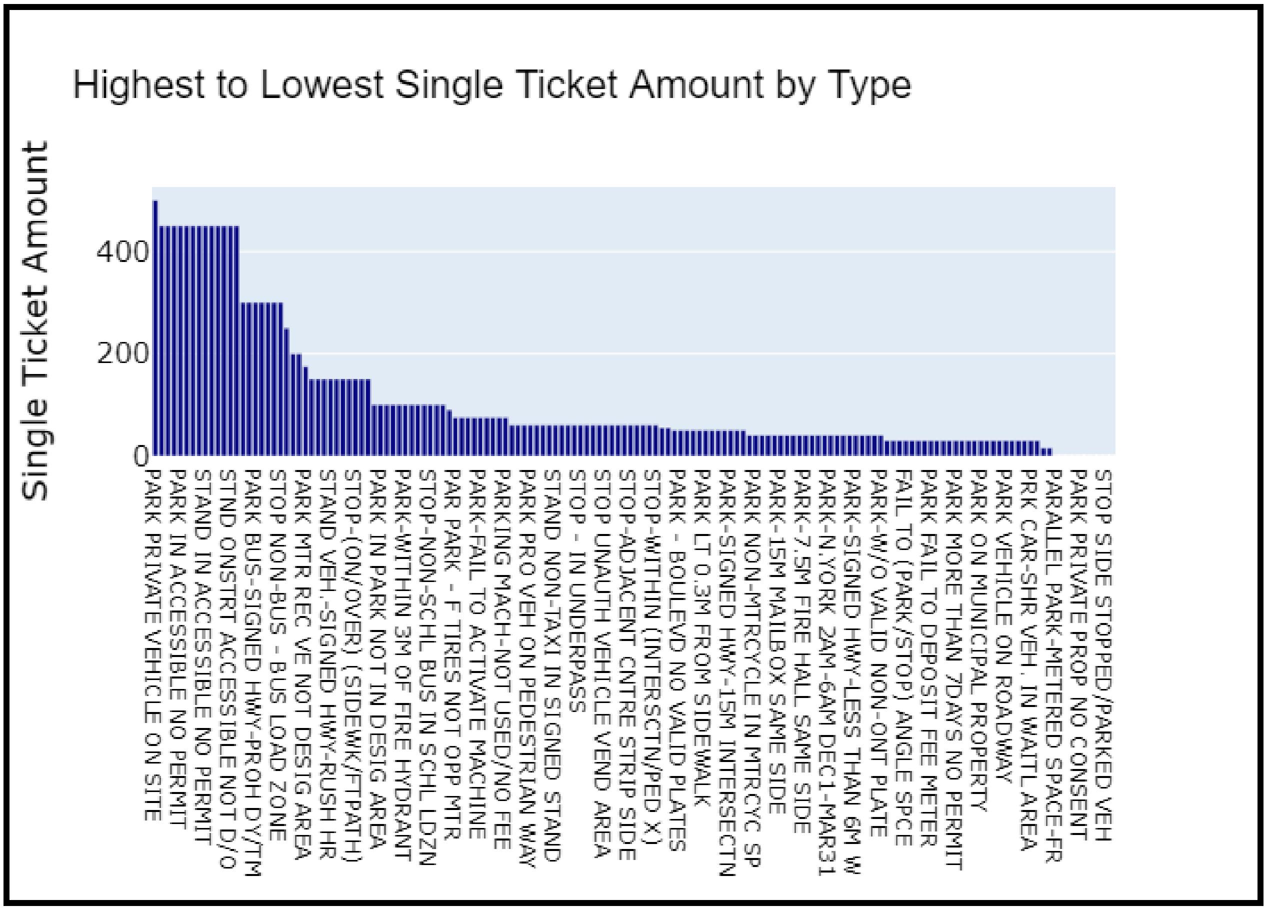


Fig 7: Highest to Lowest Single Ticket Amount By Type

This graph is a horizontal bar chart titled "Highest to Lowest Single Ticket Amount by Type." It appears to represent various types of tickets (presumably for violations or infractions of some kind) on the horizontal axis and their corresponding highest single ticket amounts on the vertical axis. Each bar represents a different type of ticket, and the length of the bar indicates the highest amount that a single ticket for that type costs. The graph shows that a few types of tickets have a significantly higher maximum cost compared to the others, as seen by the taller bars on the left. As you move towards the right, the amounts decrease. The highest fine is for parking private vehicles on site which is 500 and least being Parallel parked meter space.

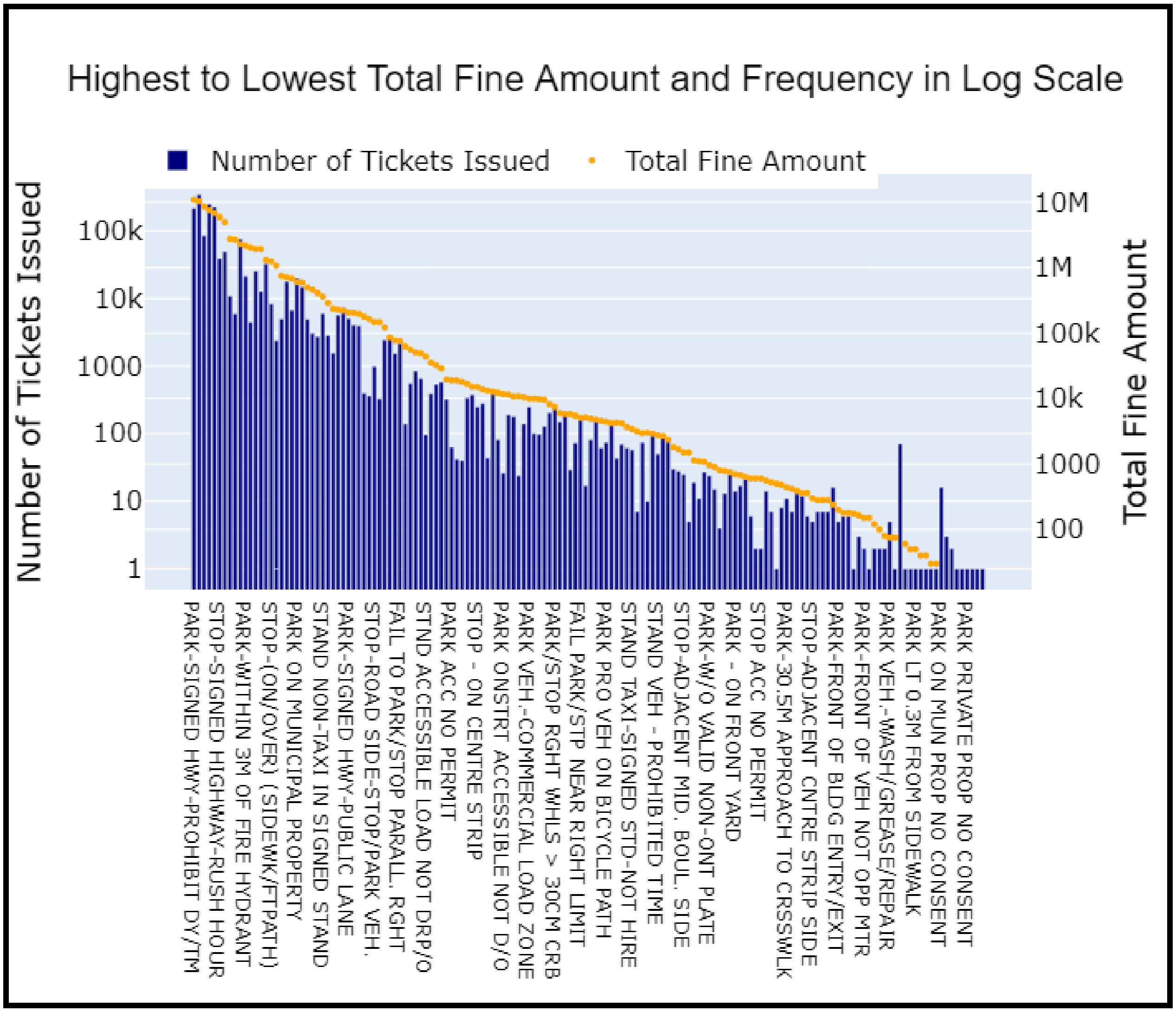


Fig 8: Highest to Lowest Total Fine Amount and Frequency in Log Scale

This graph provides a comprehensive overview of the frequency and total fine amounts associated with different violations or infractions, presented in a descending order from highest to lowest. The y-axis is on a logarithmic scale, allowing for the visualization of data points that span a wide range of values. The blue bars represent the number of tickets issued for each violation type, while the orange dots indicate the corresponding total fine amount. The x-axis displays the violation descriptions, which are abbreviated due to space constraints. At the top of the graph, we can see that the violation with the highest frequency of tickets issued and the highest total fine amount is related to something like "failure to wear seatbelts" or a similar traffic violation. This violation stands out significantly from the rest, suggesting that it is a common occurrence and a major source of revenue from fines. As we move down the graph, the frequency of tickets issued and the total fine amounts gradually decrease, with some violations having relatively high ticket counts but lower total fine amounts, and others displaying the opposite pattern. Notably, there is a long tail of violations with relatively low ticket frequencies and total fine amounts, indicating a wide range of less common infractions or violations that contribute to the overall ticketing and fine collection process. The use of a logarithmic scale on the y-axis allows for the representation of both very high and very low values within the same graph, enabling a clear visualization of the relative magnitudes and distributions of ticket frequencies and fine amounts across different violation types.

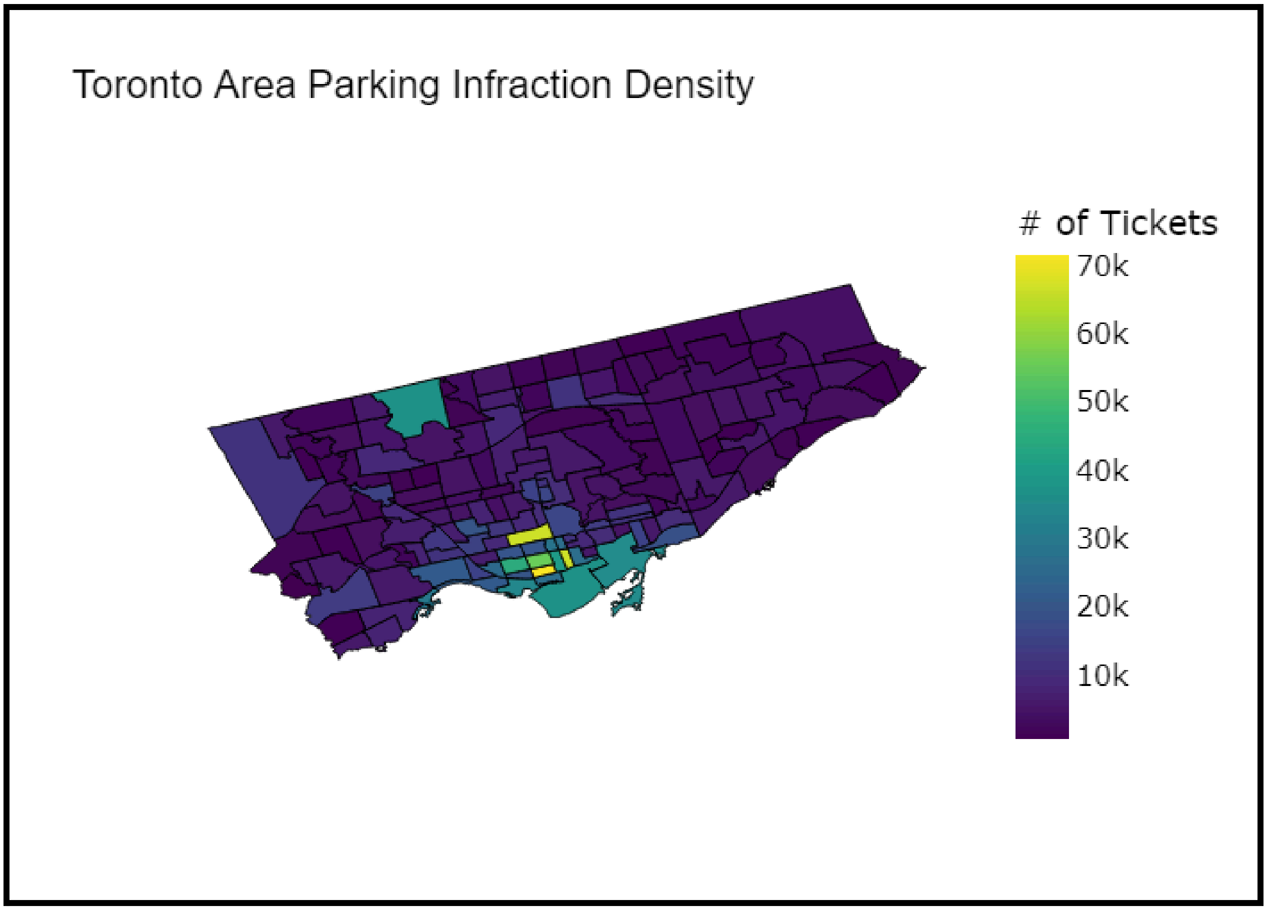


Fig 9: Toronto Area Parking Infraction Density

The map is segmented into various areas, likely representing different neighborhoods or zones within Toronto. The color scale on the right indicates the number of tickets, with darker colors representing higher numbers of tickets. For example, dark purple represents areas with the lowest density of tickets (10k), while bright yellow represents areas with the highest density (70k). The map shows that the majority of areas have a lower ticket density, while there are a few specific areas with significantly higher ticket densities, as indicated by the yellow and light green colors. These could be areas with stricter parking enforcement, higher traffic, or fewer available parking spots. This kind of visualization is helpful for identifying hotspots for parking infractions and could be used by city planners and law enforcement to analyze and address parking issues.

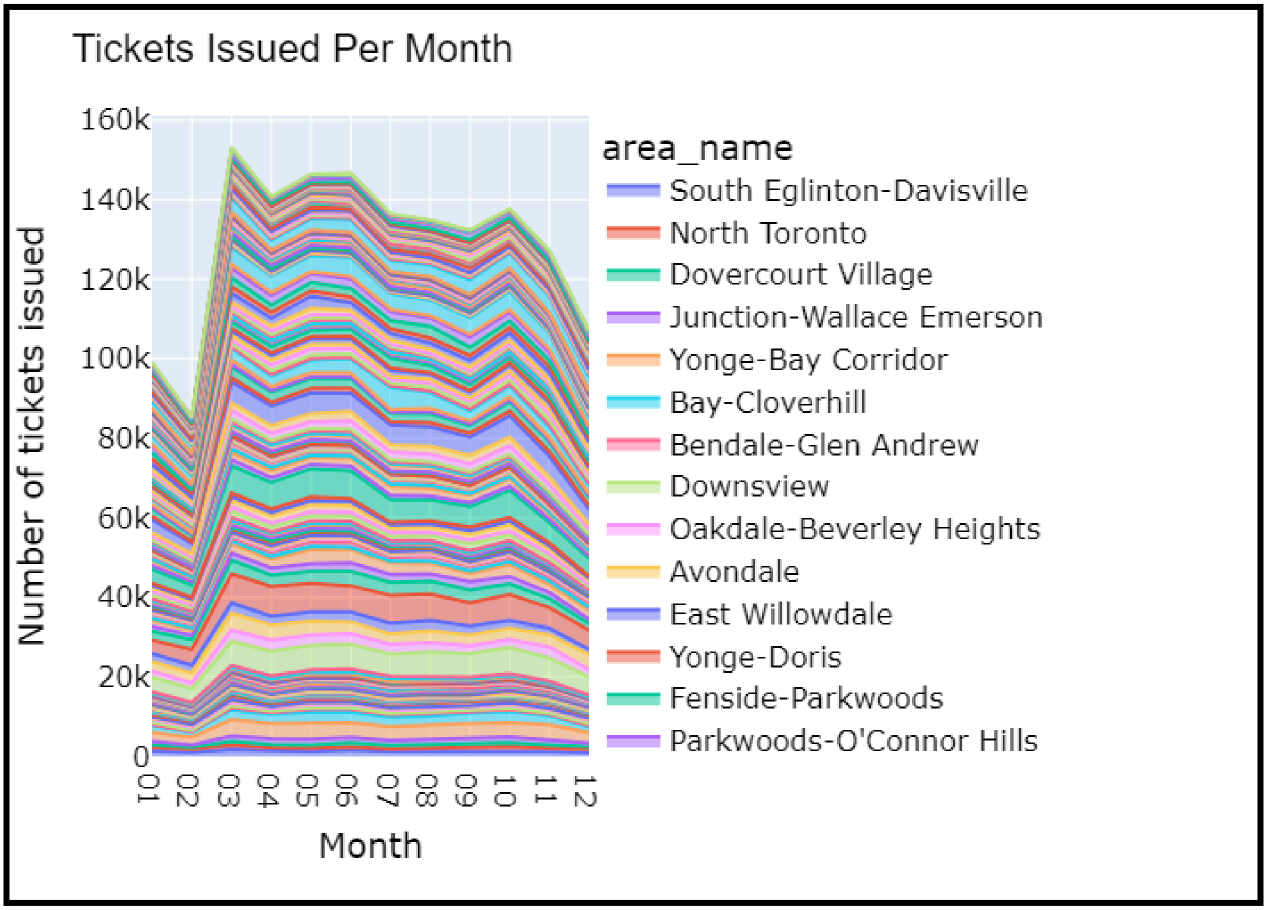


Fig 10: Tickets Issued Per Month

The graphs are Area based plots. The x-axis of the graph is labeled "Month" and appears to show the last 12 months. The y-axis is labeled "Number of tickets issued" and the scale goes from 0 to 160,000. There are twelve areas listed on the graph. It appears that the area with the most tickets issued is Downsview, with a peak of around 140,000 tickets issued in a single month. The area with the fewest tickets issued is Parkwoods-O'Connor Hills, which never appears to have had more than 12,000 tickets issued in a single month according to the graph.

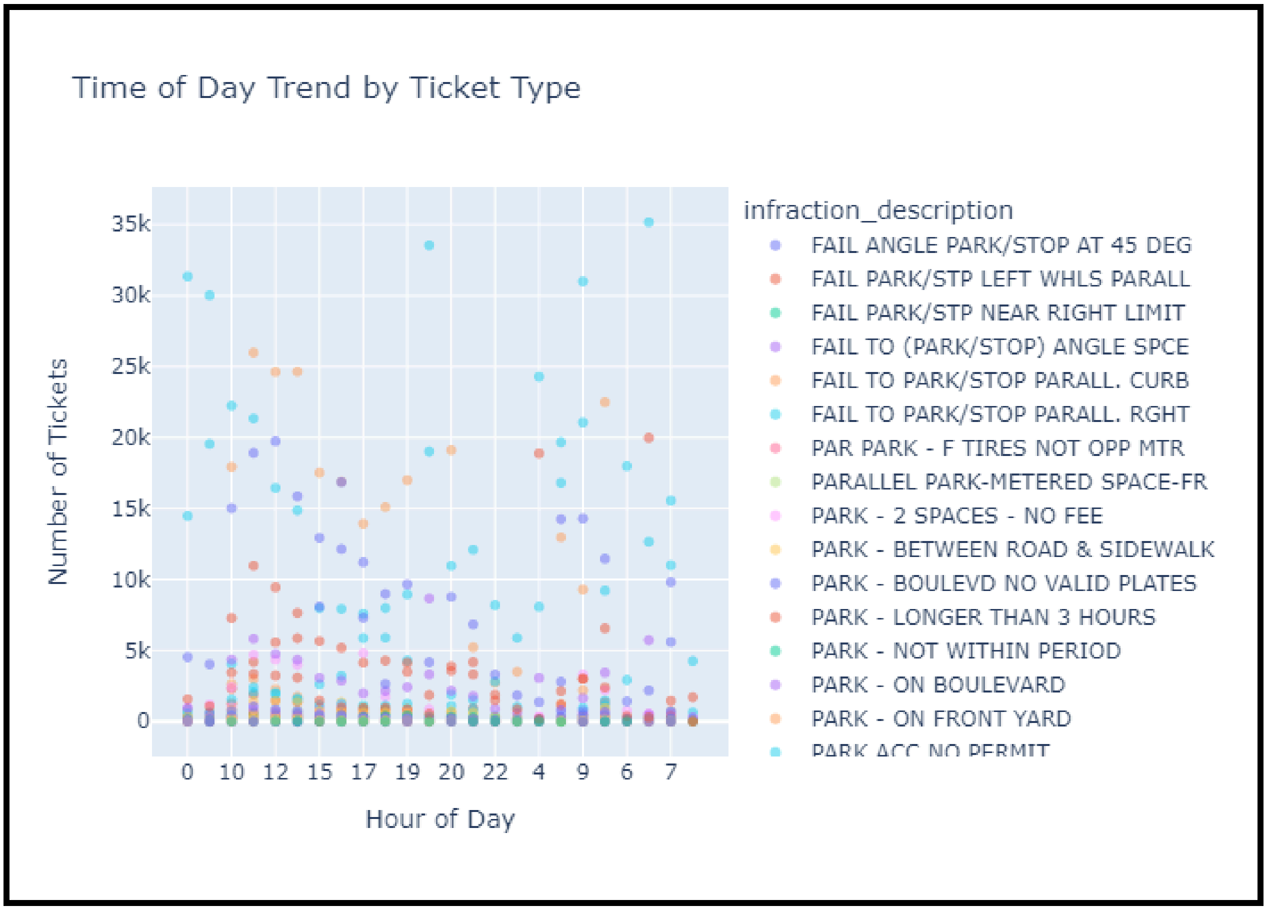


Fig 11: Time of Day Trend by Ticket Type

The graph is a scatter plot titled "Time of Day Trend by Ticket Type," which represents the distribution of various parking infractions over different hours of the day. Each dot on the plot corresponds to a specific number of tickets issued for a particular infraction at a certain hour of the day. The horizontal axis shows the hour of the day, ranging from 0 to 23, indicating a 24-hour format. The vertical axis shows the number of tickets issued, ranging from 0 to approximately 35,000. The different colors and shapes of the dots represent different types of parking infractions, as listed in the legend under the title "infraction_description." For instance, a pink circle represents "FAIL TO PARK/STOP PARALL. CURB," while a dark blue diamond represents "PARK - NO FRONT YARD."

Certain infractions are more prevalent at specific times of day. For example, some types might peak during business hours, while others could be more common in the evening or early morning. The distribution of tickets for each infraction type over the hours of the day can indicate patterns, such as higher infractions in the morning or evening rush hours. Some infraction types have a wide scatter across the day, indicating they occur frequently at various times, while others may be more concentrated at certain hours.

The data points seem to be somewhat evenly distributed across the hours for many infraction types, indicating that these infractions occur regularly throughout the day.

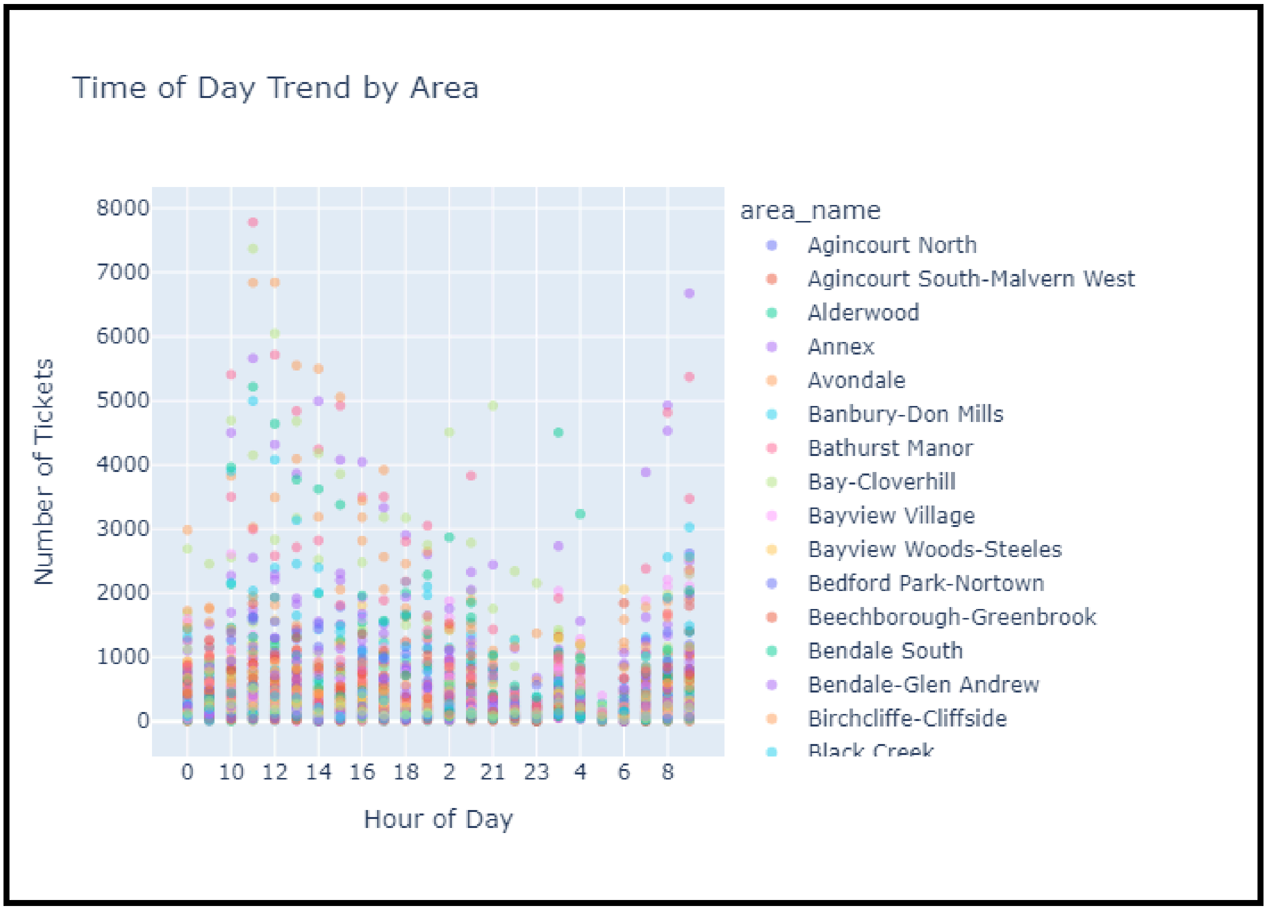


Fig 12: Time of Day Trend by Area

The graph is a scatter plot titled "Time of Day Trend by Area," displaying the distribution of ticket issuances across different times of the day for various areas. The horizontal axis indicates the time of day, using a 24-hour format from 0 (midnight) to 23 (11 PM). The vertical axis represents the number of tickets issued, which ranges from 0 to 8000. Each colored dot represents a different area within a city, presumably Toronto given the names like "Agincourt" and "Alderwood," which are known areas in Toronto. The dots are plotted based on the number of tickets issued in that area at each hour of the day. There are spikes in ticket numbers at certain times, which may correspond with either rush hours or times when parking restrictions begin or end. The distribution and density of tickets vary by area, with some areas showing a higher concentration of tickets issued at certain times of the day. No single area appears to dominate the ticketing at all hours, suggesting that enforcement or violations are spread out across the city rather than concentrated in one area. This type of visualization is useful for urban planning and law enforcement to identify when and where parking violations are most common.

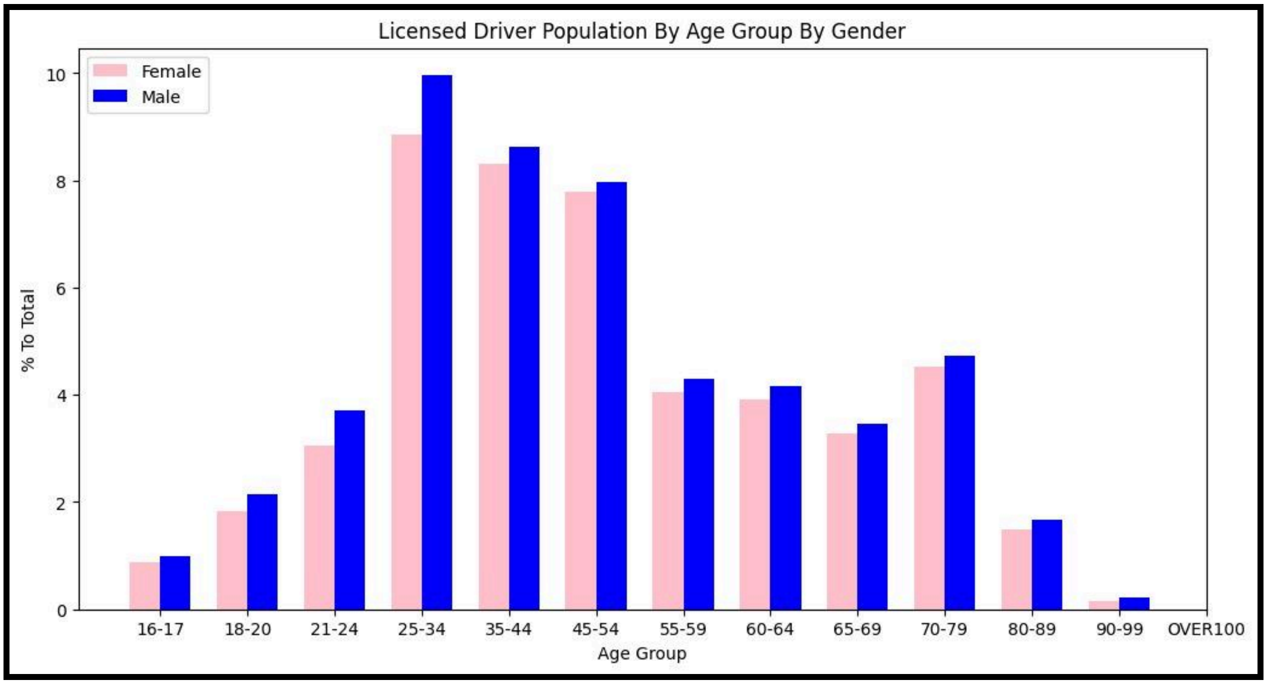


Fig 13: Time of Day Trend by Area

The graph you sent me shows the licensed driver population by age group and gender. The y-axis shows the number of licensed drivers, and the x-axis shows the age group. The data is divided into two groups, male and female. Overall, there are more male drivers than female drivers across all age groups. The highest number of licensed drivers are between the ages of 35-44.

Male: The number of male licensed drivers starts to increase significantly at 18-20 years old, and then continues to increase steadily until it reaches a peak at 35-44 years old. After that, the number of male drivers starts to decrease gradually.

Female: The number of female licensed drivers follows a similar pattern to male drivers, but with fewer drivers overall. The number of female licensed drivers starts to increase significantly at 18-20 years old, and then continues to increase steadily until it reaches a peak at 35-44 years old. After that, the number of female drivers started to decrease gradually. During 55 - 64, both genders were similar respectively during that age.

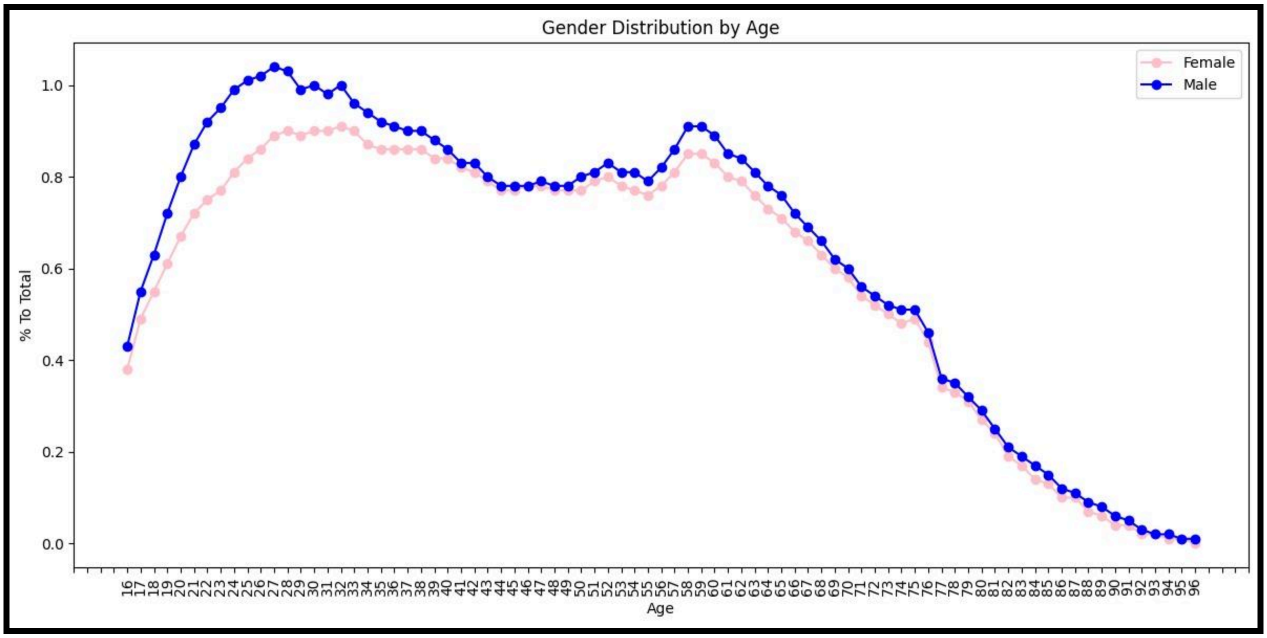


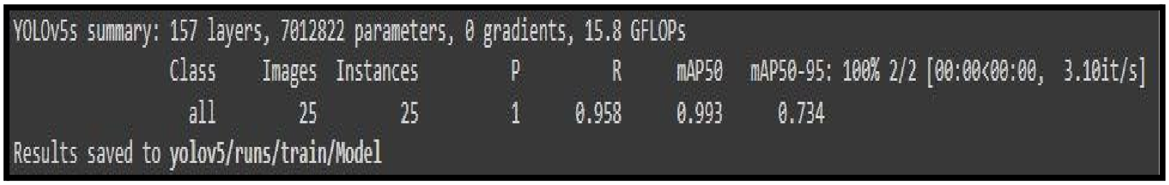
Fig 14: Time of Day Trend by Area

The graph you sent me is titled "Gender Distribution by Age". It shows the proportion of females and males in the United States population according to age. The x-axis of the graph is labeled "Age" and goes from 0 to 100. The y-axis is labeled "% To Total" and goes from 0% to 100%. There are two lines on the graph, one colored blue and one colored pink. The legend at the top of the graph indicates that the blue line represents females and the pink line represents males. The graph shows that there are more females than males in the United States population from birth to about age 50. From about age 50 onwards, the proportion of males in the population starts to increase and eventually becomes higher than the proportion of females. The gap between the proportion of females and males in the population starts to narrow around age 1840 to 50. Then there is a small spike, but both genders merge later at the age of 60.

Our solution effectively addresses two critical tasks: license plate recognition and driver drowsiness detection. To achieve precise license plate recognition, we opted for the pre-trained YOLO model and then retrained it on a dataset that includes vehicles images with their number plates. As license plates are a specialized category we used Transfer Learning. YOLO is renowned for its accuracy and real-time processing capabilities. By dividing images into a grid system, YOLO efficiently detects objects, ensuring both accuracy and speed, which outperforms existing models plagued by low precision and slow processing. Hence, a pre-trained YOLO model might have some capability to detect vehicles, it will not have been specifically trained for number plate detection unless it has been fine-tuned on an appropriate dataset containing labeled images of number plates.

For detecting driver drowsiness, we employed a CNN architecture due to its ability to extract essential features from raw image data, including subtle facial expressions and eye movements. This sophisticated approach ensures robust performance, even in varying real-world conditions, offering prompt feedback on driver states with high accuracy. The scalability of our solution allows for further optimization, promising enhanced road safety through vigilant monitoring of driver alertness.

<u>For License plate recognition:</u>



```
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 2/2 [00:00<00:00,  3.10it/s]
                   all         25         25          1      0.958      0.993      0.734
Results saved to yolov5/runs/train/Model
```

Fig 15: YOLO Summary

Using the YOLO, results seem to indicate that precision, which measures the accuracy of the model's predictions, is perfect at 1, indicating that whenever the model identifies an object, it's almost always correct. Recall, which assesses how well the model captures all instances of the objects present, is quite high at 0.958, suggesting that the model does a good job of finding most of the objects in the images. The mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5 is impressively high at 0.993, indicating the model's ability to precisely locate objects without much overlap. However, the overall mAP score of 0.734, calculated across various IoU thresholds, suggests that the model may struggle a bit more when objects overlap or are closely packed together.

<u>For Driver Drowsiness:</u>

Our CNN model performance was tracked during different phases. Then we plotted the accuracy and loss from both the training and validation datasets.

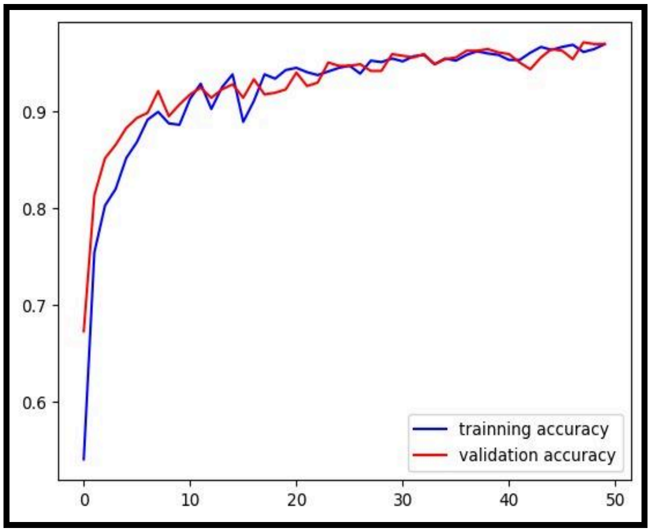


Fig 16: Accuracy Plot

The Accuracy Graph depicts a significant surge in performance initially, with both training and validation accuracy steadily increasing, indicating notable improvements. However, a subsequent plateau suggests that the model has likely reached its maximum accuracy potential given the current data and architecture. We have achieved an accuracy of 0.9689. Additionally, the convergence of training and validation accuracy implies balanced generalization without overfitting.

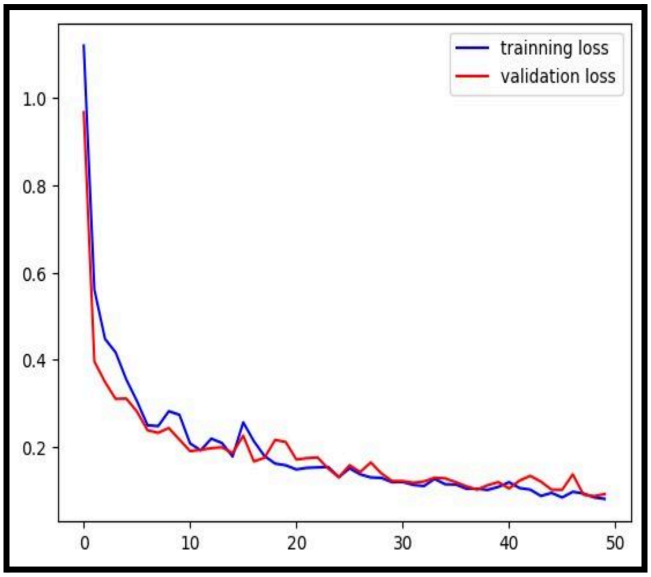
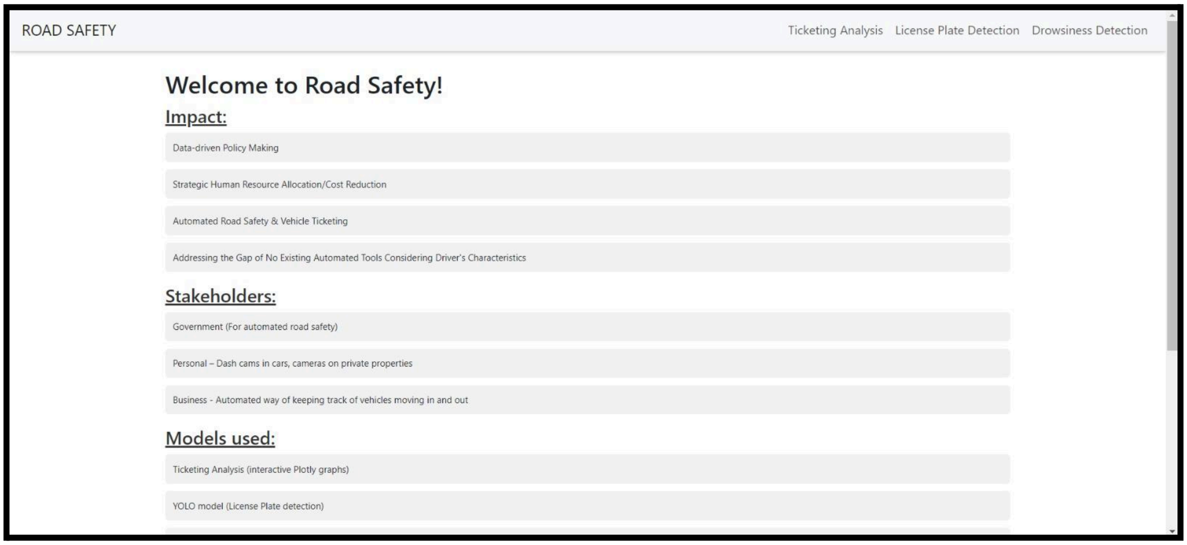


Fig 17: Loss Plot

In contrast, the Loss Graph demonstrates a sharp decline in training and validation loss, followed by a gradual leveling off. This pattern reflects the model's rapid error reduction early in training, stabilizing as it approaches optimal performance. The minimal difference between training and validation loss, along with their parallel trajectories, suggests the model is not overfitting and is effectively learning generalized patterns applicable to unseen data.

Hence, The model exhibits strong convergence, characterized by the stability of accuracy and loss metrics. Both training and validation curves closely track each other, indicating the model's robust generalization capability. There are no indications of overfitting, as evidenced by the simultaneous decrease in validation loss alongside training loss, and the close alignment of validation accuracy with training accuracy.

Data Product


ROAD SAFETY

Ticketing Analysis License Plate Detection Drowsiness Detection

Welcome to Road Safety!

Impact:

Data-driven Policy Making

Strategic Human Resource Allocation/Cost Reduction

Automated Road Safety & Vehicle Ticketing

Addressing the Gap of No Existing Automated Tools Considering Driver's Characteristics

Stakeholders:

Government (For automated road safety)

Personal – Dash cams in cars, cameras on private properties

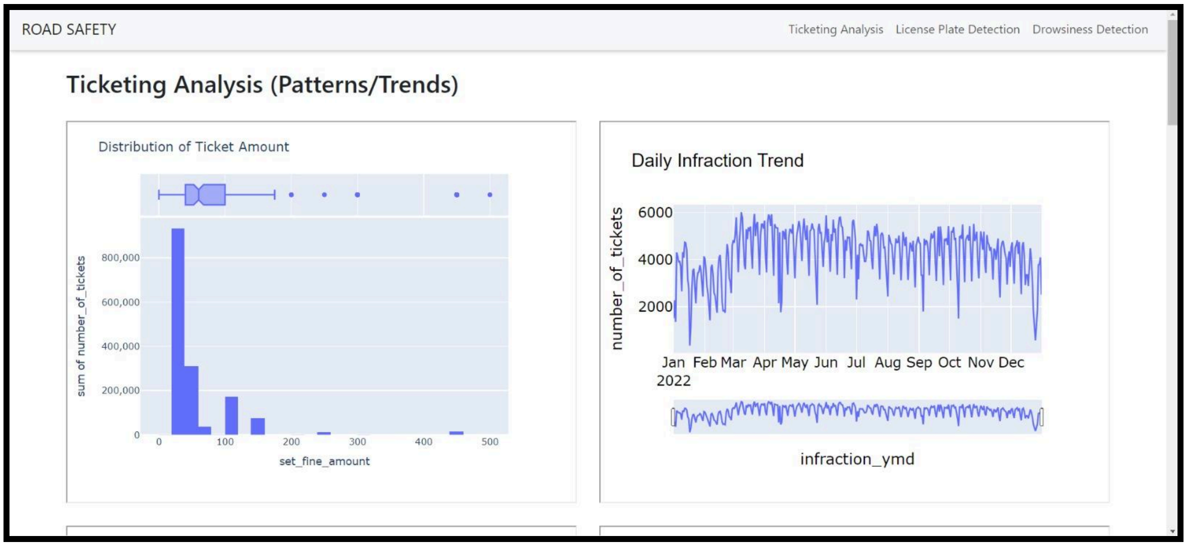
Business - Automated way of keeping track of vehicles moving in and out

Models used:

Ticketing Analysis (interactive Plotly graphs)

YOLO model (License Plate detection)

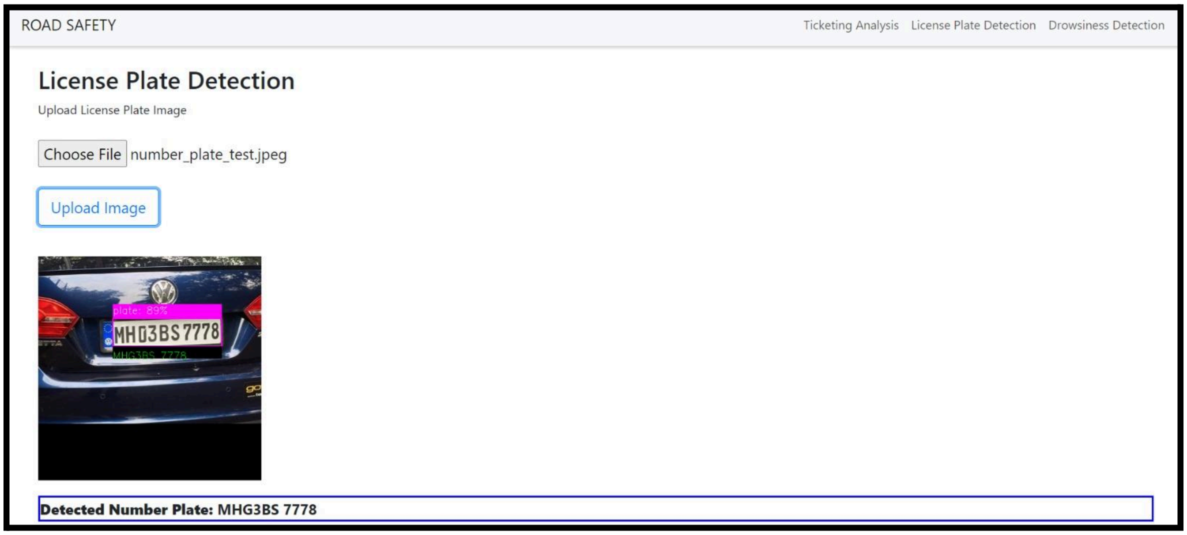
Fig 18: Main Page


ROAD SAFETY

Ticketing Analysis License Plate Detection Drowsiness Detection

Ticketing Analysis (Patterns/Trends)

Fig 19: Ticketing Analysis Page


ROAD SAFETY

Ticketing Analysis License Plate Detection Drowsiness Detection

License Plate Detection

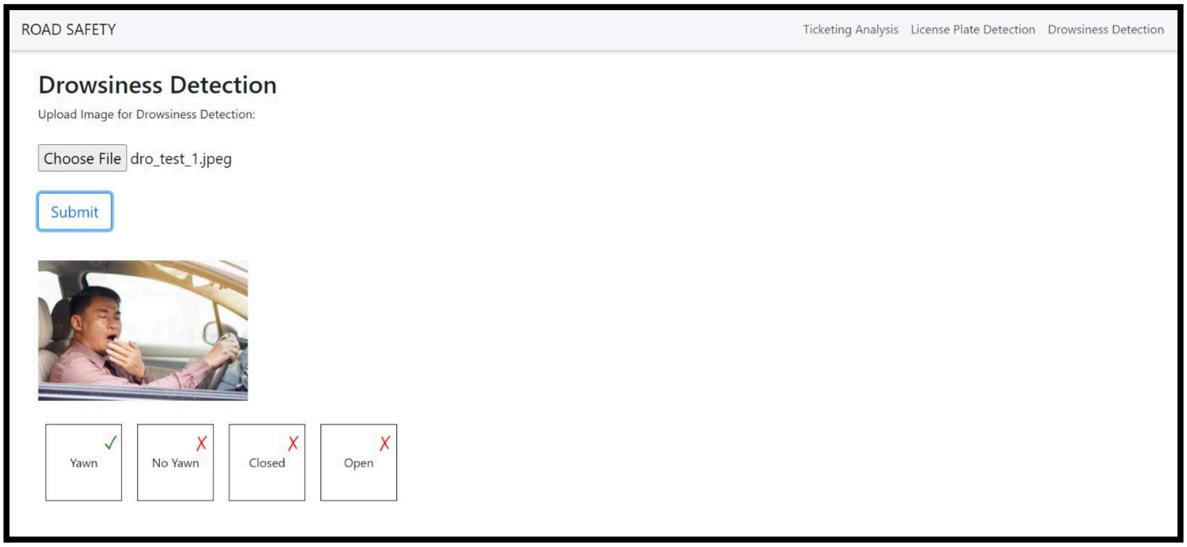
Upload License Plate Image

Choose File number_plate_test.jpeg

Upload Image

Detected Number Plate: MHG3BS 7778

Fig 20: License Plate Detection Page


ROAD SAFETY

Ticketing Analysis License Plate Detection Drowsiness Detection

Drowsiness Detection

Upload Image for Drowsiness Detection:

Choose File dro_test_1.jpeg

Submit

Yawn ✓

No Yawn ✗

Closed ✗

Open ✗

Fig 21: Drowsiness Detection Page

Youtube Link: https://youtu.be/wa_gxY8OSuE

Our data product is a dynamic Dashboard App built using Flask, featuring a user-friendly interface crafted using HTML, CSS, and Bootstrap for frontend design, and leveraging JavaScript and Flask for backend functionality. The Dashboard boasts a Navigation bar offering convenient navigation to three distinct pages: Ticketing Analysis, Driver Drowsiness, and License Plate.

The Ticketing Analysis Page hosts captivating interactive graphs, offering insights based on various segments and criteria. Users can engage with these graphs using interactive options that appear upon hover.

On the Driver Drowsiness and License Plate pages, users encounter an upload feature enabling them to submit images. Once submitted, the backend machine learning models swing into action, providing prompt results such as detection of Yawn or No Yawn, as well as indications of Open or Closed for Driver Drowsiness, and extraction of text for License Plate recognition.

Lessons Learned

- ❖ Gathering datasets from diverse sources, employing techniques such as pdf scraping and data collection, before merging them to suit our specific requirements. Our merging process was paramount, heavily reliant on accurately determining whether the ticketing locations fell within the polygon coordinates delineating the boundaries of the neighborhood regions. Once merged, we proceeded with a meticulous series of steps encompassing data cleaning, processing, integration, feature engineering, and visualization of the combined dataset. Utilizing tools such as matplotlib and plotly, we crafted visually informative graphs, ensuring interactive usability for enhanced insights.

- ❖ In the realm of model development, we dedicated efforts to training and fine-tuning our models tailored to our unique datasets, aiming to enhance accuracy. Opting for a YOLO model proved advantageous over crafting our own from scratch, considering the significant time investment required for training. This decision enabled us to streamline the development process and focus on optimizing model performance for our specific use case.

- ❖ Through this project, we gained comprehensive experience in building web applications from the ground up, encompassing both frontend and backend development. This involved crafting intuitive user interfaces using HTML, CSS, Bootstrap, and JavaScript, while simultaneously structuring the backend with Flask to efficiently handle requests, process data, and deliver results.

- ❖ Furthermore, we acquired proficiency in seamlessly integrating machine learning models into the application, empowering real-time analysis and predictions based on user input. This required a deep understanding of backend architecture to effectively manage model interactions and ensure smooth performance.

- ❖ Throughout the development process, we encountered numerous challenges that demanded creative problem-solving and troubleshooting skills. From debugging code to optimizing performance and addressing compatibility issues across various browsers and devices, we honed our ability to overcome obstacles and deliver a polished, functional product.

- ❖ Strategizing, organizing, and adhering to project timelines were pivotal aspects of our project management approach, ensuring timely completion while juggling various tasks such as assignments and quizzes. Despite encountering setbacks such as bugs, we proactively planned for contingencies, allocating additional time in advance to accommodate unforeseen challenges. This proactive approach enabled us to navigate obstacles effectively and maintain momentum towards achieving project objectives within the designated time frame.
- ❖ In essence, this project provided us with a holistic understanding of Data gathering, machine learning integration, data visualization, web application development, project management, and the intricacies of problem-solving in a real-world context. These skills are not only applicable to this project but also transferable to a wide range of domains within software engineering and data science.

Summary

The project represents a multifaceted exploration into enhancing urban transportation safety and efficiency through the integration of machine learning and data analysis techniques. It centers around three pivotal areas: drowsiness detection, license plate recognition, and ticketing analysis.

In drowsiness detection, the project delves into machine learning algorithms to mitigate the risks posed by drowsy driving. By analyzing physiological signals and behavioral data, the project aims to develop robust models capable of accurately identifying patterns indicative of driver drowsiness. Through the implementation of face detection algorithms and convolutional neural networks (CNNs), the project seeks to provide timely alerts to drivers, thus preventing potential accidents.

Automated license plate recognition (ALPR) systems play a crucial role in traffic rule enforcement. The project harnesses machine learning algorithms to enhance the efficiency and accuracy of ALPR systems by precisely extracting and recognizing license plate numbers from surveillance camera images. Through the utilization of advanced models such as YOLO, the project endeavors to improve feature extraction and processing speed, thereby optimizing license plate detection and recognition capabilities.

Ticketing analysis is driven by the imperative to enhance road safety and traffic management. By integrating diverse datasets encompassing parking ticket records, socio-economic data, and driver demographics, the project aims to uncover underlying trends and patterns in ticketing. Through the application of advanced statistical methodologies, the project seeks to identify key factors contributing to parking violations, thus informing strategic resource allocation and optimizing enforcement efforts.

The project follows a meticulously crafted data science pipeline, encompassing comprehensive stages such as data collection, exploration, cleaning, integration, analysis, and model development. Leveraging a suite of tools including Python, Pandas, Plotly, Matplotlib, Tabula, CNN model, YOLO model, and Flask, the project addresses specific challenges encountered at each stage of the pipeline.

Evaluation of the solution demonstrates the efficacy of the developed machine learning models in achieving precise license plate recognition and drowsiness detection. The culmination of the project is a dynamic and interactive data product in the form of a dashboard application. Integrating machine learning models for real-time analysis and prediction, the dashboard provides users with actionable insights and information, thus empowering stakeholders to make informed decisions and enhance road safety.

Through the project journey, valuable lessons are gleaned in project management, data acquisition, machine learning model development, web application deployment, and problem-solving. These lessons contribute to a comprehensive understanding of real-world applications in urban transportation and data science domains.

Overall, the project represents a significant endeavor to revolutionize urban transportation safety and efficiency through innovative applications of machine learning and data analysis techniques. It stands as a testament to the potential of data-driven approaches in addressing complex challenges and driving positive societal impact.

Future Scope

In the future, to further enhance the efficiency and effectiveness of our road safety solution, we aim to build upon the following ideas and explore additional avenues:

❖ Weather Conditions Prediction: Integrate weather data into your system to predict road conditions in real-time. This can include features like rain prediction to alert drivers about slippery roads, fog prediction to advise caution, and snow prediction to suggest alternative routes or slower driving speeds.

❖ Pedestrian Interaction Analysis: Develop a model that can detect and analyze interactions between vehicles and pedestrians. This can include identifying pedestrian crossings, predicting pedestrian movements to warn drivers, and integrating with traffic signals to enhance pedestrian safety.

❖ Emergency Response Integration: Collaborate with emergency response services to integrate our system with their operations. This can include automatic alerts to emergency services in case of accidents, providing real-time data about road conditions to emergency vehicles, and optimizing response times to improve overall road safety.

References

1) https://ieeexplore.ieee.org/document/9182237
2) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7917813/
3) https://ieeexplore.ieee.org/abstract/document/8343528
4) https://ieeexplore.ieee.org/document/10071305