CS418 Final Project Report

Project Title: Project WalkSafe – GitHub Repo

Final Presentation: Google Slides

Date: 05/04/2025

Group Members: Sammy Haddad, Ameer Mustafa, Alaa Musa, Shahriar Namvar, Javid Uddin

Description

Project WalkSafe is a data science project focused on making cities safer for pedestrians. Using real crash data from Chicago, we analyzed when and where pedestrian accidents are most likely to occur. We used Python and mapping tools to clean and filter data, find patterns, perform EDA, and create multiple visualizations to highlight possible trends and correlations. Finally, we performed ML analyses to identify pedestrian crash hotspots in different neighborhoods, determine the best features that impact pedestrian incidents, and analyze whether speed cameras were an effective traffic device tool for reducing pedestrian crash incidents. We used various models and techniques, mainly from packages such as scikit-learn and SciPy. The goal was to turn hefty raw data into something useful and draw potential conclusions and insights to help city planners, researchers, or even everyday people make smarter, safer decisions about where they walk.

Data

Note: These datasets are too large to submit here, so we included links to them instead!

1. Chicago Speed Camera Violations
2. Chicago Traffic Crash Data
3. Chicago Traffic Tracker Estimates

When working with this dataset, many of our ML and visuals focused on pedestrian and pedalcyclist crashes, so we filtered it through taking a subset of the DataFrame like below. Additionally, for any spatial visualizations / ML analyses, we also combined this with Chicago neighborhoods using GeoPandas (bottom half of code snippet).

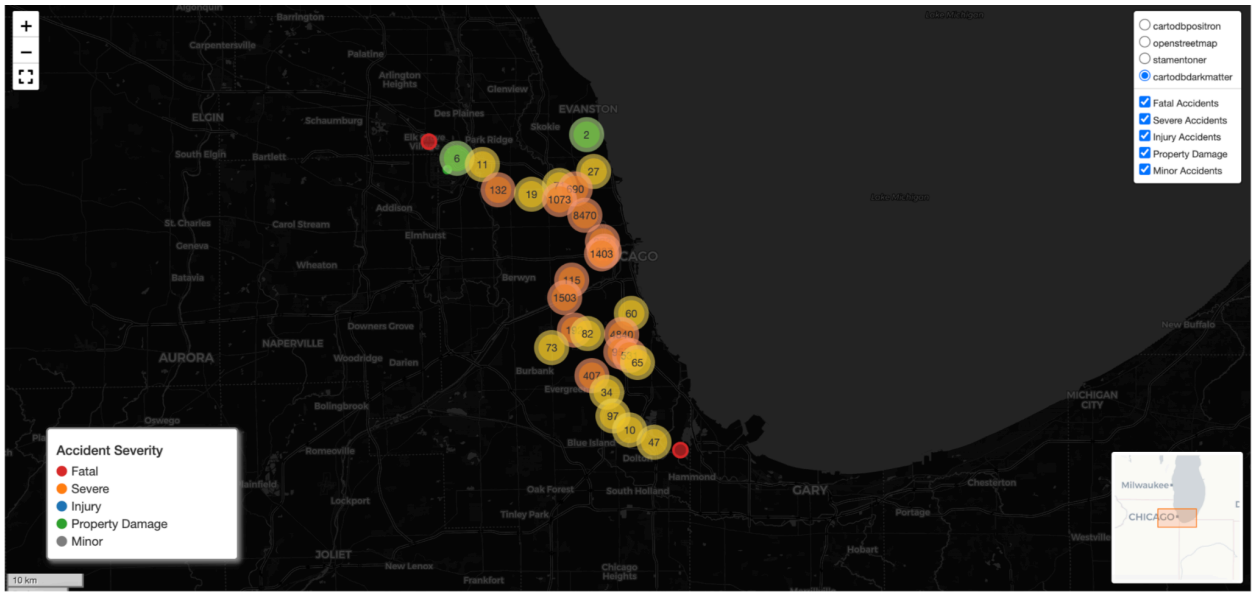
```
traf_crash_df = pd.read_csv('traffic_crashes.csv')
# ~ 40,000 accidents that were pedestrian related incidents
traf_crash_df[(traf_crash_df['FIRST_CRASH_TYPE'] == 'PEDESTRIAN') |
(traf_crash_df['FIRST_CRASH_TYPE'] == 'PEDALCYCLIST')].head(5)
# Create the dataset into geo dataframe using latitude and longitude columns
geometry = [Point(xy) for xy in zip(traf_crash_df['LONGITUDE'],
traf_crash_df['LATITUDE'])]
crash_gdf = gpd.GeoDataFrame(traf_crash_df, geometry=geometry, crs=4326)
# Read and display the chicago neighborhood geojson file
chicago_gdf = gpd.read_file('chicago.geojson')
# Joining the crashes dataset with the chicago neighborhood geojson file
chi_crashes_gdf = gpd.sjoin(crash_gdf, chicago_gdf, predicate='within')
```

For cleaning the datasets, we created and used the following definitions below.

```
def load_data():
   # Load crashes data with proper datetime parsing
   crashes = pd.read_csv("traffic_crashes.csv")
   crashes.columns = crashes.columns.str.strip()
   crashes['CRASH_DATE'] = pd.to_datetime(crashes['CRASH_DATE'], format='%m/%d/%Y
%I:%M:%S %p', errors='coerce')

   # Load speed camera data
   cameras = pd.read_csv("speed_camera_violations.csv", parse_dates=['VIOLATION
DATE'])
   cameras.columns = cameras.columns.str.strip()

   # Load traffic data
   traffic = pd.read_csv("chicago_traffic_tracker.csv", parse_dates=[' LAST_UPDATED'])
   traffic.columns = traffic.columns.str.strip()
```

```
    return crashes, cameras, traffic

def clean_data(crashes, cameras, traffic):
    # Clean crashes data
    crashes = crashes.dropna(subset=['LATITUDE', 'LONGITUDE'])
    crashes['POSTED_SPEED_LIMIT'] = pd.to_numeric(crashes['POSTED_SPEED_LIMIT'],
errors='coerce')

    # Clean camera data
    cameras = cameras.dropna(subset=['LATITUDE', 'LONGITUDE'])
    cameras['VIOLATIONS'] = pd.to_numeric(cameras['VIOLATIONS'], errors='coerce')

    # Clean traffic data
    traffic = traffic[traffic['CURRENT_SPEED'] > 0]  # Remove invalid speed records

    return crashes, cameras, traffic
```

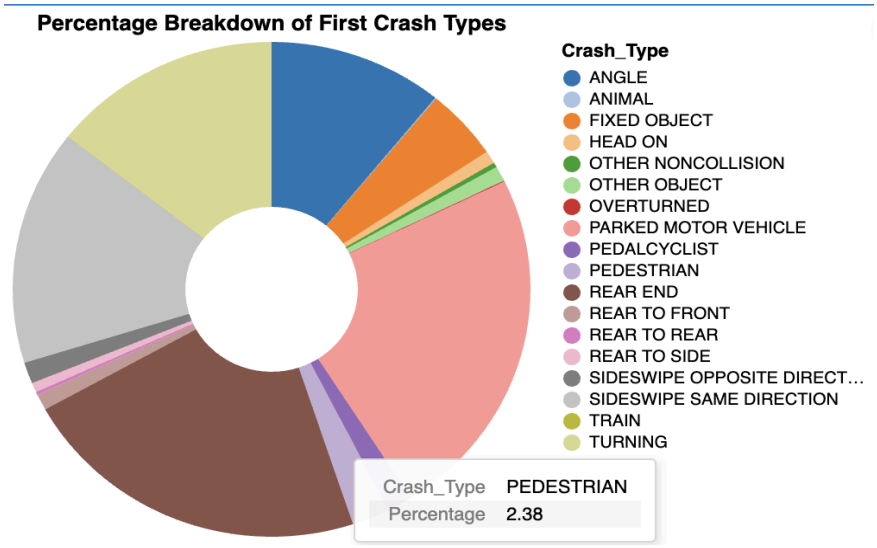
Additional Work: Interactive Map - Alaa Musa



The interactive map displays pedestrian-involved traffic crashes in Chicago, broken down by severity level (Fatal, Severe, Injury, Property Damage, and Minor). Data-driven insights can be obtained, such as connections between weather, time of day, and crash severity. Additionally, the map's ability to show harmful areas, patterns, and trends helps to enhance safety by highlighting high-risk regions for infrastructure upgrades (such as speed bumps, traffic signals, and crosswalks) and raising public awareness of pedestrian accident risk zones. The date, injuries, speed limit, and address are shown next to each collision when a marker is clicked. These markers are grouped into clusters to decrease clutter and improve the map's aesthetic appeal. As you zoom in, clusters start to show up more frequently. To allow for faster map navigation, a minimap is incorporated. Additionally included are a marker severity filter and a map skin changer. Lastly, each severity level's significance is explained in a legend.

Visualizations

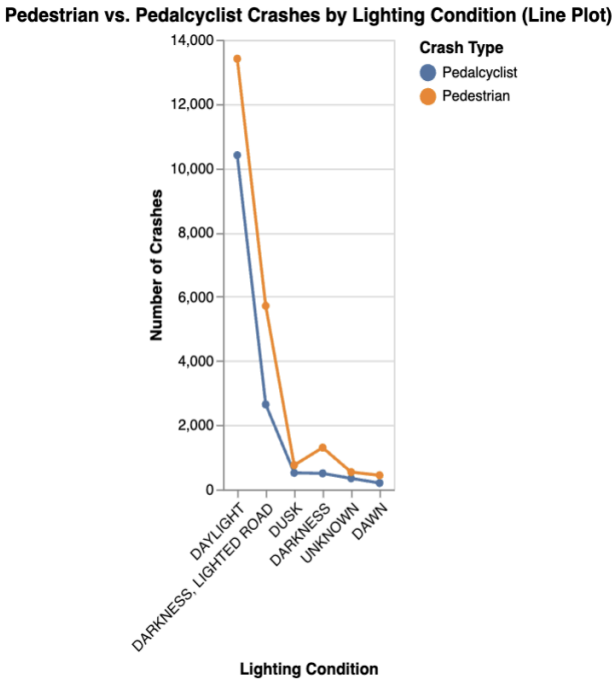
Visual #1 - Crash Type Distribution - Ameer Mustafa

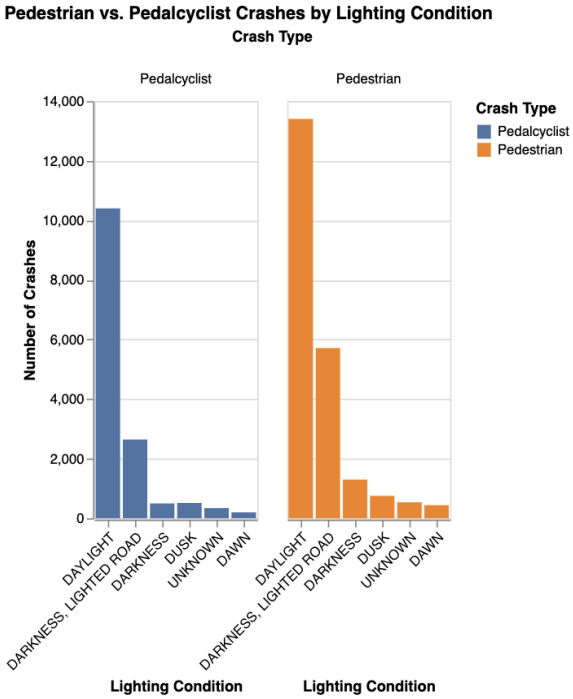


Explanation:

The Pie Chart was created utilizing the Chicago Traffic Crash Dataset. The inspiration behind the visualization was to be able to easily depict the different types of crashes that are contained in the database. The chart revealed there are a total of 18 crash types. We are mainly interested in exploring the Pedestrian and Pedalcyclists incidents since those are the only two that directly involve pedestrians. Pedestrian-type crashes account for ~2.5% and Pedalcyclist for ~1.5% bringing our total exploration dataset to 4% of the original database. By filtering the dataset to include only these crash types, we reduce the data from over 1 million rows to approximately 40,000, making analysis more manageable and computationally efficient.

Visual #2 & #3 - Lighting Effects on Pedestrian-Related Crashes - Ameer Mustafa

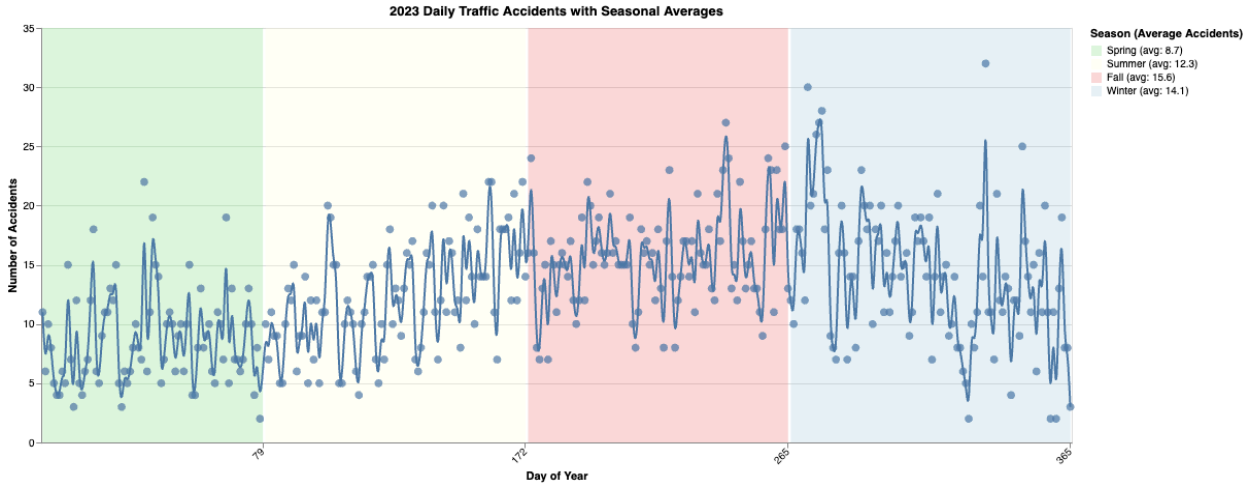




Explanation:

We wanted to explore how lighting conditions impacted pedestrian-related incidents. Therefore, we filtered the data based on pedal-cyclist and pedestrian separately and plotted them side-by-side utilizing a bar chart to see if we could find any correlations. The data revealed that the outcome is consistent across both domains. Daylight and Darkness Lighted Road tend to be the biggest two lighting conditions in which most crashes occur, with daylight being the highest by far. Additionally, I decided to use a line plot to compare them on the same graph to visualize the same data from a different perspective. The results of both graphs are shown above.

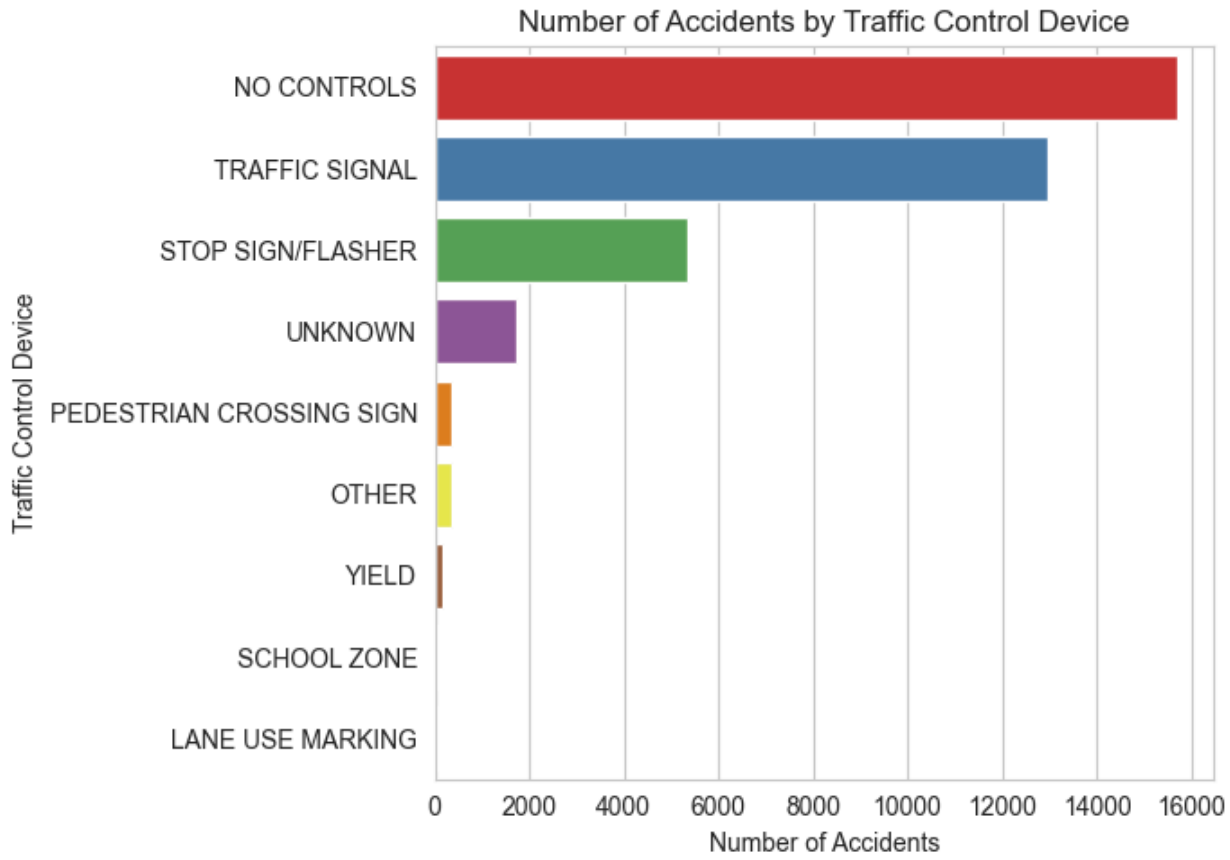
Visual #4 - Seasonal Trends in Pedestrian-Related Crashes - Ameer Mustafa



Explanation:

In order to explore the seasonal impacts on pedestrian-related crashes, we decided to create a scatter plot with the addition of a best-fit line. This graph was created utilizing the filtered Chicago traffic crash dataset. A few key insights derived were that Fall had the highest average daily number of pedestrian-related incidents in 2023. This result was counterintuitive since we originally anticipated winter to be the highest due to poor weather conditions. A possible reason for this outcome is that drivers tend to get overconfident when weather conditions are good, leading them to pay less attention and be more prone to accidents. As opposed to winter, they are fully aware that their vehicle can lose control at any given moment, naturally leading them to be more aware of their surroundings.

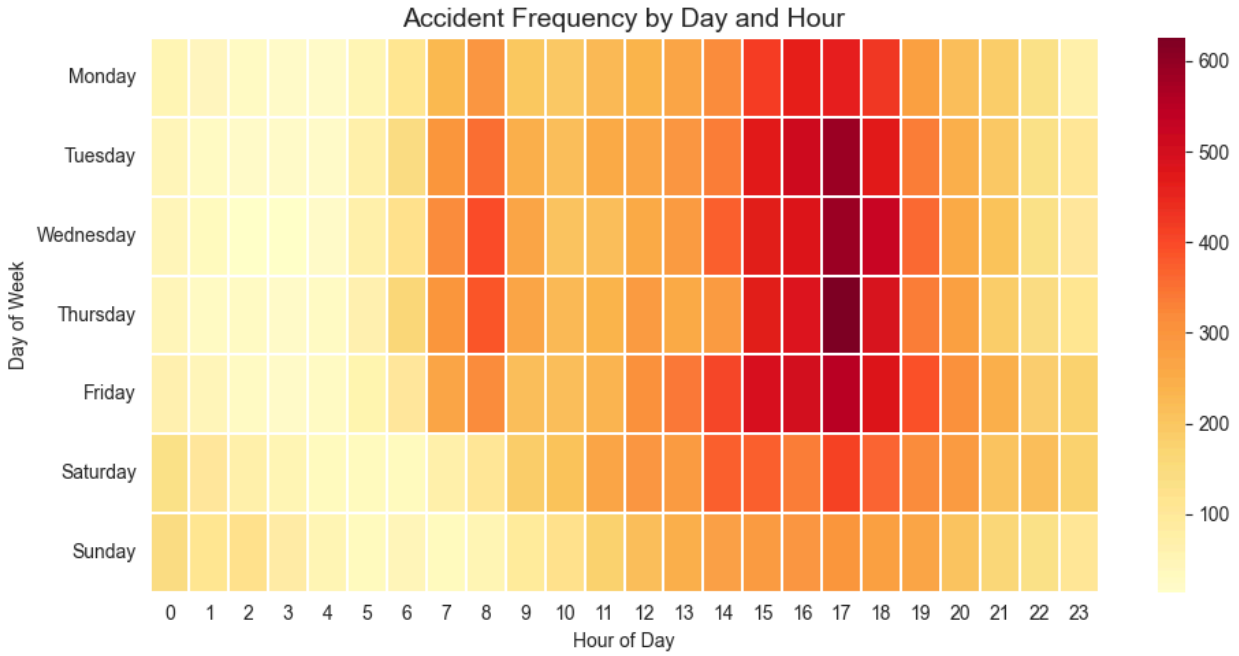
Visual #5 - Shahriar Namvar



Explanation:

This plot shows the number of traffic accidents per type of traffic control device present at the location of the incident. The data demonstrates that the highest number of accidents occurred in areas without any traffic control devices. This suggests a strong correlation between the absence of traffic control systems and the likelihood of accidents, emphasizing the need for better infrastructure in such areas. Traffic signals account for the second-highest number of accidents, with more than 12,000 incidents, followed by stop signs or flashers at around 6,000. This indicates that even in places with control devices, driver behavior such as failure to obey signals at intersections remains a significant contributor to accidents.

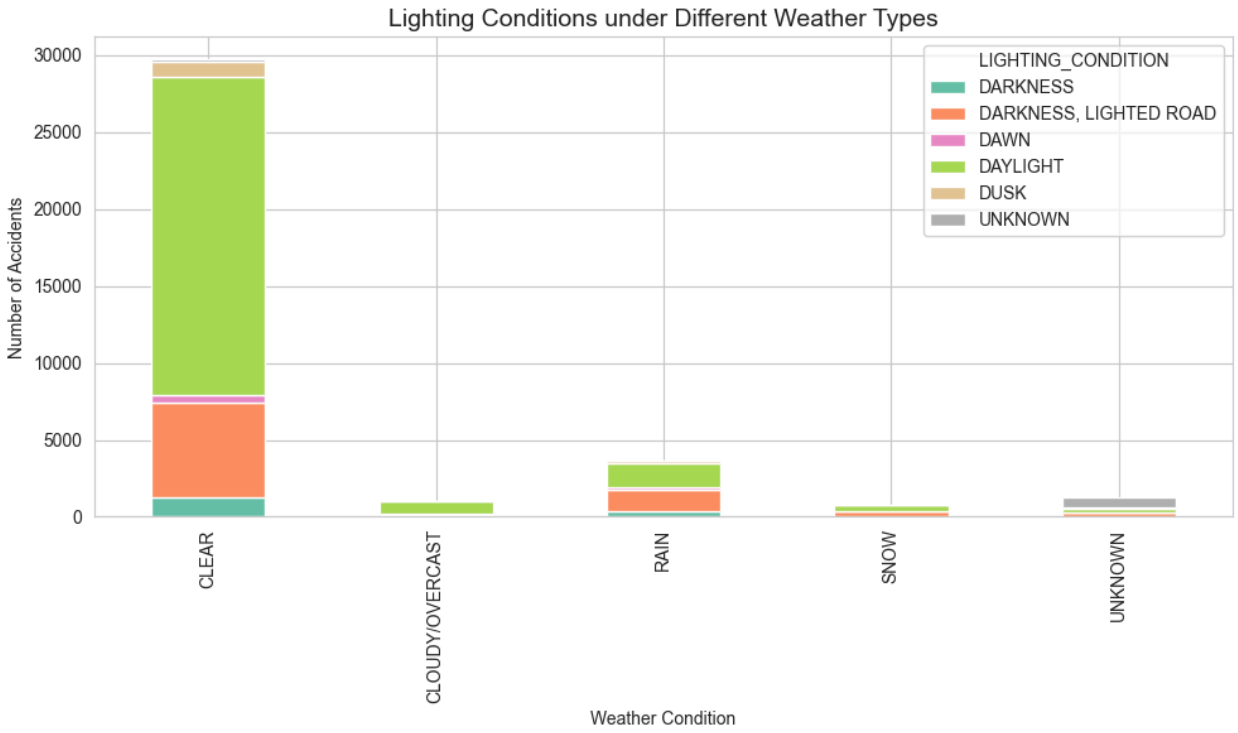
Visual #6 - Shahriar Namvar



Explanation:

The heatmap shows the number of accidents by the hour of the day across each day of the week. The pattern shows that the peak number of accidents occurs during the afternoon hours between 3 PM and 6 PM from Monday to Friday. This aligns closely with rush hour, when congestion is at its highest. Tuesday through Friday account for the majority of accidents while weekends show a relatively lower number of accidents, specifically during early morning hours. This temporal analysis provides valuable insights into critical periods with high accident risk and highlights time periods where implementing targeted safety measures, such as increased traffic enforcement or proper safety measures could be most effective in reducing incidents.

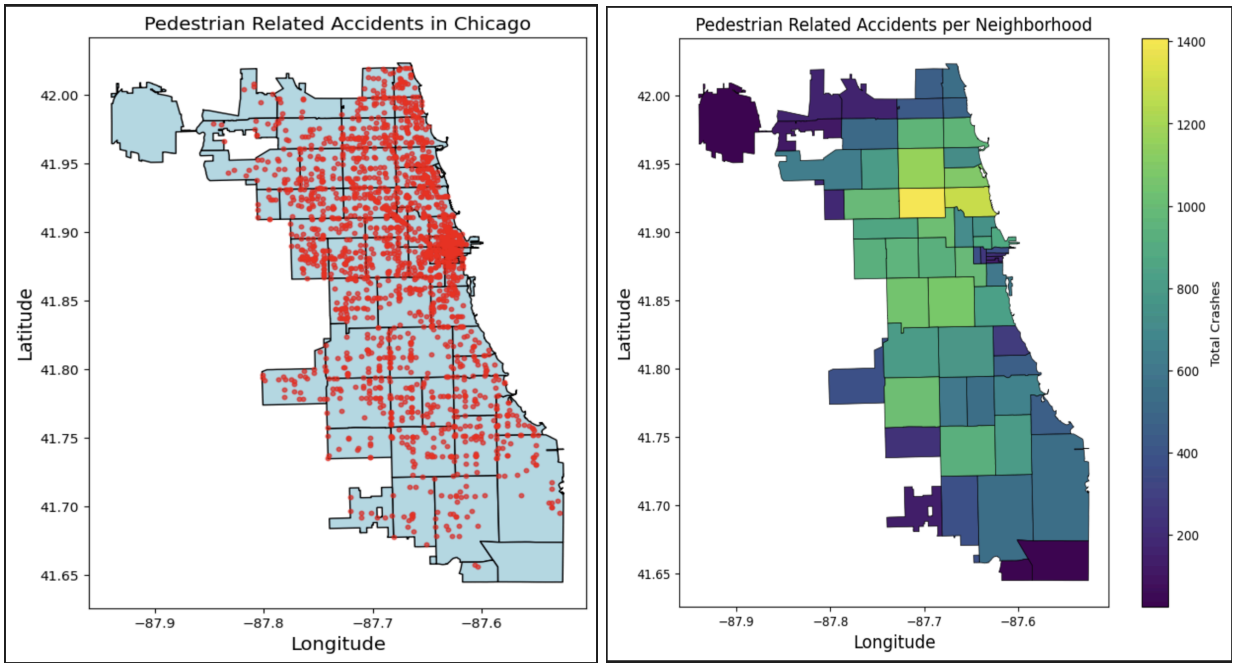
Visual #7 - Shahriar Namvar



Explanation:

This stacked bar chart explores how lighting and weather conditions impact the frequency of accidents. Surprisingly, the majority of accidents happen under clear weather conditions, particularly in daylight. This contradicts our initial hypothesis and implies that poor weather conditions such as rain and snow can not necessarily increase the number of accidents. Although accidents can occur, they are comparatively less common in rainy, snowy, or low-light situations like dusk or darkness. In fact, poor weather conditions might cause the drivers to be more cautious and careful while driving.

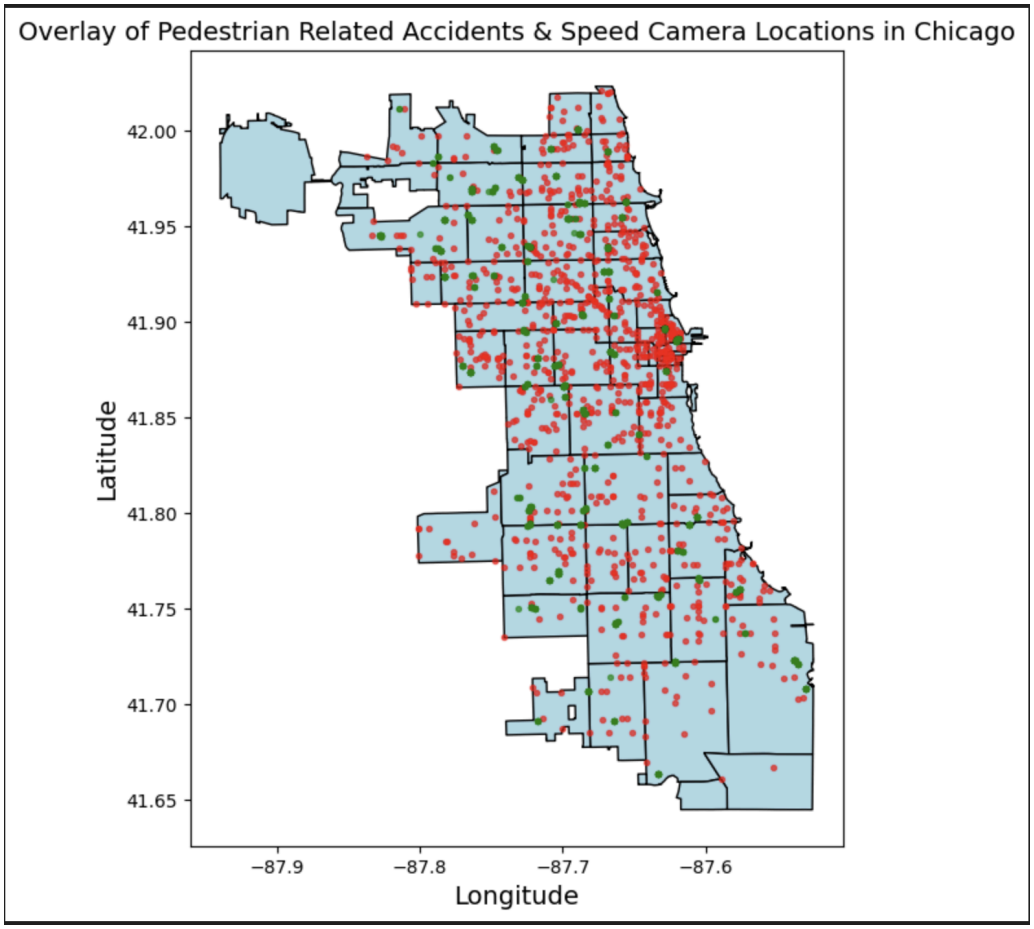
Visual #8 & #9 - Maps of Pedestrian Related Incidents across Chicago - Sammy Haddad



Explanation:

The above maps are a dot density (left) and choropleth map (right), showcasing the spatial distribution of pedestrian related incidents in the city of Chicago and how pedestrian incidents compare across neighborhoods. For the dot density map, due to the fact that the dataset is large (~930k crashes and ~37k pedestrian related ones), we took a sample of this to make the plot a bit more readable and display where the most dense areas were in terms of pedestrian crashes. For the choropleth map, we get a better idea of which neighborhoods are producing a high number of pedestrian related accidents, meaning when employing traffic safety measures these are the areas that should be prioritized first.

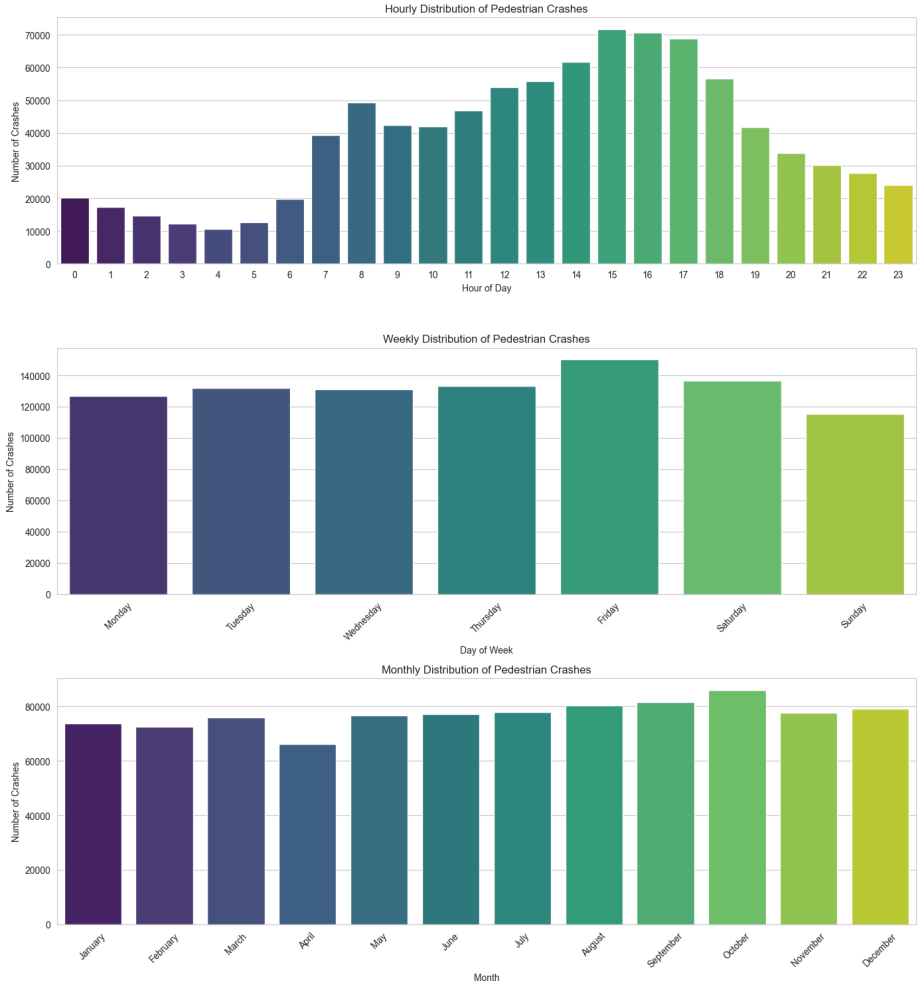
Visual #10 - Overlay of Speed Cameras and Pedestrian Incidents - Sammy Haddad



Explanation:

The above dot density map showcases the spatial distribution of pedestrian related incidents per neighborhood in Chicago and overlays this the speed camera locations. To create this plot, we took a sample of both of these datasets since plotting all speed camera locations and pedestrian incidents would be far too cluttered and not readable! This map allows us to get an idea of the effectiveness of speed cameras in reducing pedestrian incidents, as we can see whether crashes are generally close to, or far from, speed cameras. However, on its own, this map is not enough to determine the effectiveness of speed cameras fully, so in our 3rd ML Analysis we explored this more precisely with a KD Tree!

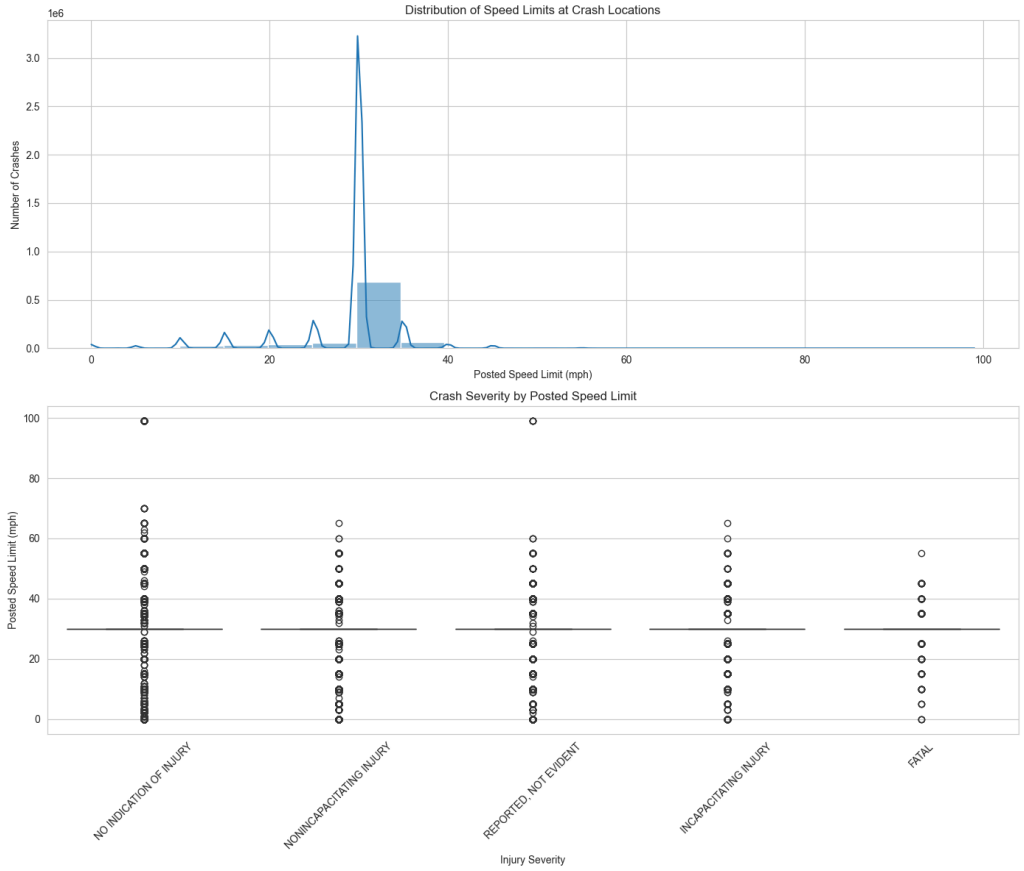
Visual #11 - *Distributions of Pedestrian Crashes* - Javid Uddin



Explanation:

This visual was created to analyze patterns in pedestrian crashes across different times of day, days of the week, and months of the year. The purpose was to identify when pedestrian incidents are most likely occurring so that we could hopefully draw further conclusions with our ML analyses. The results show a spike in crashes during late afternoon to early evening (around 3-6 PM) with Fridays being the most dangerous day of the week. October had the most traffic accidents out of all of the months, which indicates that seasonal factors may play a role.

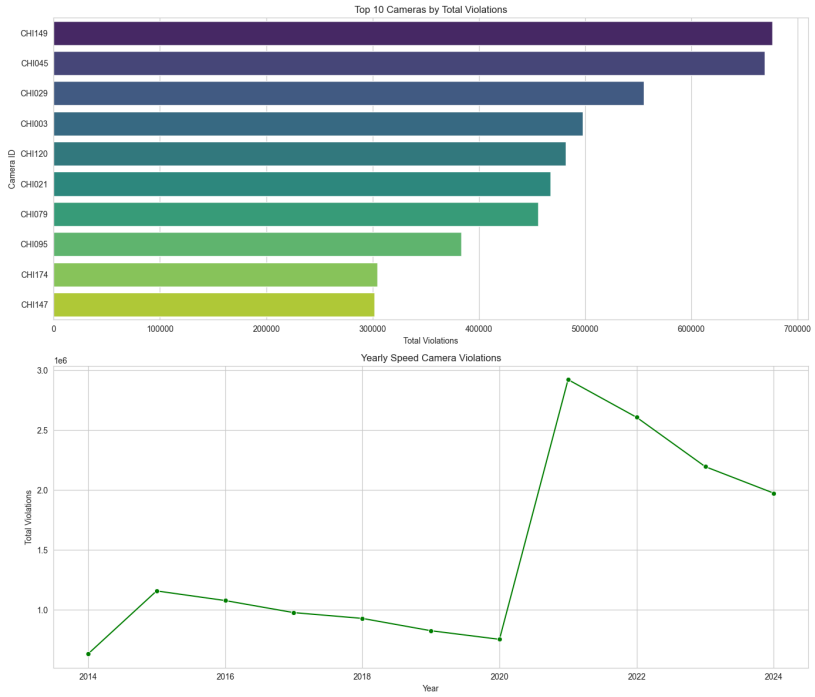
Visual #12 - *Distribution of Speed Limits at Crash Locations* - Javid Uddin



Explanation:

The visual explores how posted speed limits correlate with pedestrian crash frequency/severity. The top line plot shows that most crashes actually occur on roads with speed limits around 30mph, which is the speed limit around most of the city. Chances are that drivers involved in crashes were more than likely going well above the speed limit considering that it is low. This analysis helped motivate us to look more into speed limit cameras, and see if having them can reduce pedestrian related accidents.

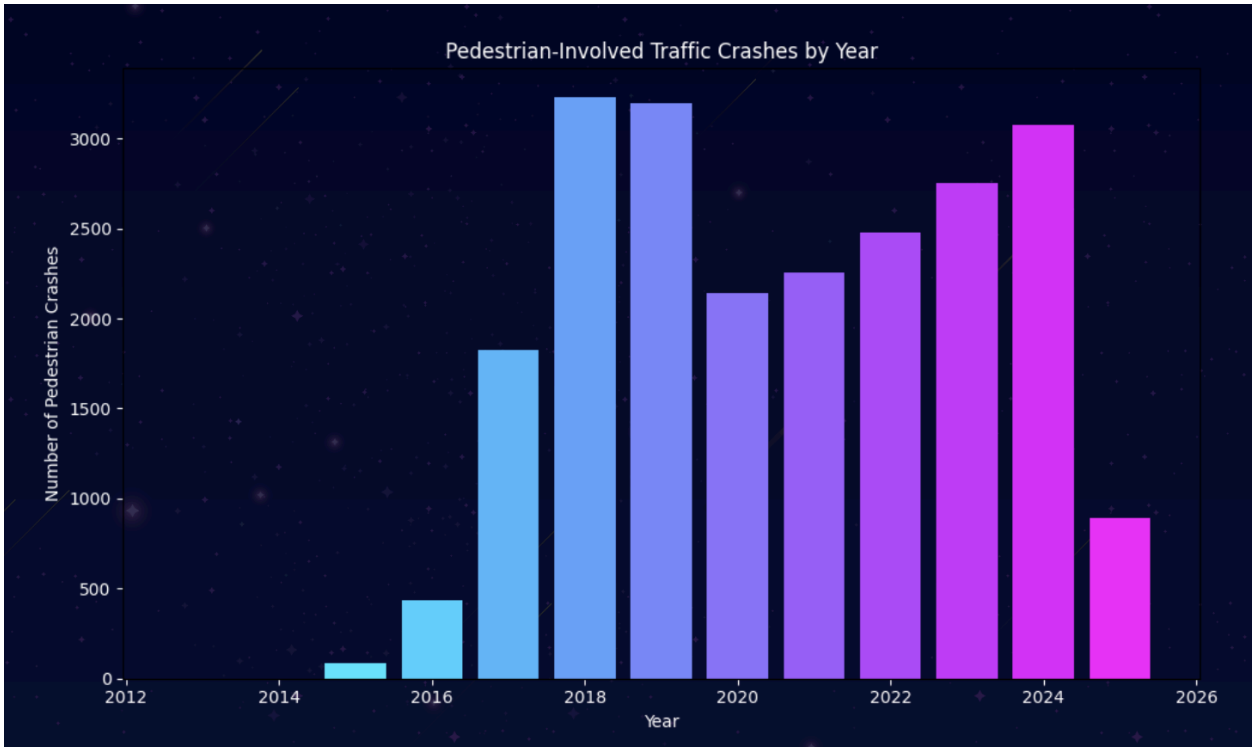
Visual #13 - *Tracking Speed Camera Violations Over Time* - Javid Uddin



Explanation:

This visual tracks speed camera violations by the top 10 cameras in Chicago, in order to understand traffic patterns and safety impacts over time (yearly). The top visual specifically showcases where speeding is most prevalent and camera enforcement is strongest. The bottom visual shows the change in speed camera violations over time, indicating a huge spike from 2020 to 2021 and therefore after. Possible causes for this were due to more people driving after covid-19, hidden speed cameras with minimal signage/warning, and carelessness by drivers (ignoring speeding tickets). This data promoted the next visual which checks to see if the number of camera violations is directly proportional to the number of pedestrian related accidents.

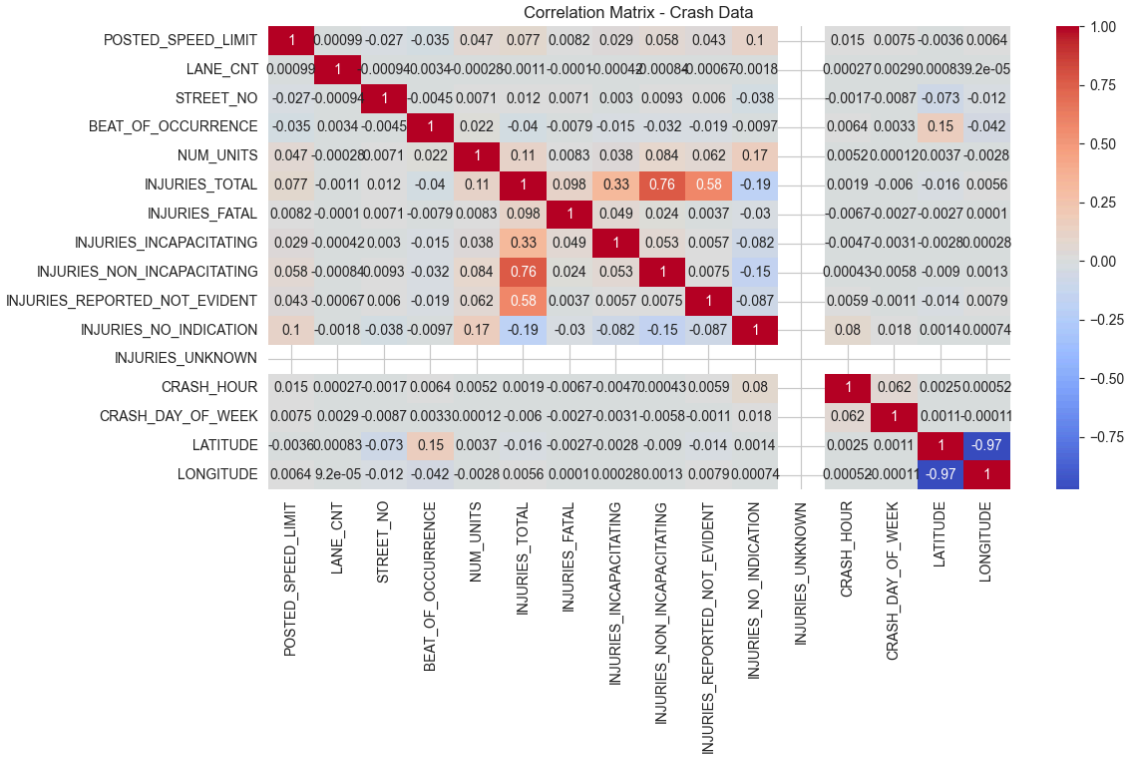
Visual #14 - *Pedestrian-Involved Traffic Crashes by Year* - Javid Uddin



Explanation:

This plot was created to visualize the yearly trend of pedestrian-involved traffic crashes, which helped us assess whether pedestrian safety has improved or worsened over time, specifically in response to speed camera violations. It turned out that our initial hypothesis with the other visual was wrong – the number of accidents also went up with the number of speed camera violations from 2020-2024 (just more consistent than the speed camera violations which had a huge spike from 2020-2021). This indicates that speed camera violations are not perfectly proportional to reducing traffic (pedestrian) related accidents.

Visual #15 - *Correlation Matrix of Crash Data* - Javid Uddin



Explanation:

This correlation matrix plot was generated to ideally show how different crash-related variables are related (statistically). It helped us identify potential patterns and predictors of crash severity. Evidently, the total number of injuries is strongly correlated with specific injuries like non-incapacitating and incapacitating injuries . Posted speed limit and total injuries had less (positive) correlation which sounds off at first but makes sense, considering that in visual #2, we saw that most crashes occur at a posted speed limit of 30mph – and since the posted speed limit in most areas is 30mph, it isn't much of a strong argument for our analyses or possible reasoning for predicting crashes (or their severity).

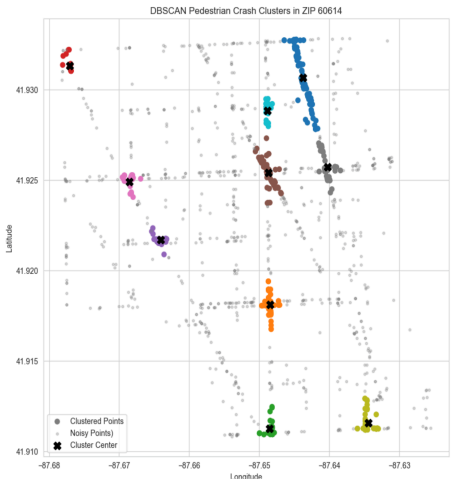
ML Analyses: Samuel Haddad

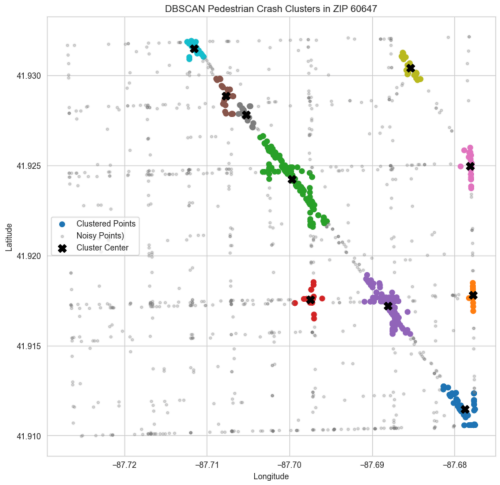
First Analysis: Identify Hotspots for Pedestrian Incidents

To begin our ML analyses, we wanted to identify, for different neighborhoods, where the hotspots of pedestrian crashes were. This is important to gather, as one of the first steps to mitigating pedestrian related incidents is figuring out where many accidents are happening, and applying the best traffic safety controls to those areas as soon as possible. To do this analysis, we used the clustering algorithm DBSCAN for different neighborhoods to identify the hotspots. We chose DBSCAN as it performs very well with spatial data that is noisy and not arranged in typical cluster shapes (since we are dealing with accident data the clusters are forming along intersections and roads, so some cross-like patterns, oddly shaped lines, etc).

Results:

Below we are showcasing the results of running DBSCAN on 2 neighborhoods with high pedestrian crash accidents compared to other neighborhoods.





From these plots we can see that DBSCAN worked well in identifying different hotspots in these neighborhoods, and allow us to have a much better idea of where to place traffic control devices and enact road safety measures to make these neighborhoods more walkable and more pedestrian friendly!

Second Analysis: Rank Feature Importance for Pedestrian Incidents

In this analysis, we want to identify which features of our dataset (the attributes / columns) are most impactful and important in determining what causes a pedestrian related incident. To do this, we will use 3 ML techniques: Logistic Regression, Random Forest, and XGBoost. We will use all of these techniques to see how they assign weights to the features, and this will clue us in to which features are most important for pedestrian related accidents. Before running any of these models, we transformed our data a bit to make it a binary classification task, so that any row that was a 'PEDESTRIAN' or 'PEDALCYCLIST' crash was given the class label 1 (pedestrian related), and all other were class 0 (non-pedestrian). With this in place, we can now showcase the most important features for each model!

2a: Logistic Regression

For our logistic regression model, we imported it using the scikit-learn Python package and fit it to our training dataset, which we created by splitting the dataset into two parts, 80% for training and 20% for testing. In the model it is important to note the class weights we assigned, and how class 1 (pedestrian related) is 18 times that of class 0 (non-pedestrian). We did this in an effort to offset the heavy class imbalance in our dataset (only 4% pedestrian related incidents). We landed on this value after testing various different values such as 10, 20, 15, and so on until we found that 18 performed best. Below is the code snippet for creating and fitting the model for our data.

```
log_regression = LogisticRegression(max_iter=1000, class_weight={0: 1, 1: 18})
log_regression.fit(X_train, y_train)
```

2b: Random Forest

For our Random Forest model, we did the same thing as above for logistic regression; used Scikit-learn packages and fit the model to our dataset based on the training set. In the model it is important to note the class weights we assigned, and how class 1 (pedestrian related) is 18 times that of class 0 (non-pedestrian). We did this for the same reason as listed above in logistic regression. Similarly, we set the max depth and minimum sample splits to be 10 to ensure that the model did not overfit the data too heavily by using too many features that were essentially irrelevant. Below is the code snippet for creating and fitting the model for our data.

```
rf = RandomForestClassifier(class_weight={0: 1, 1: 18}, n_estimators=200,
max_depth=10, min_samples_split=10)
rf.fit(X_train, y_train)
```

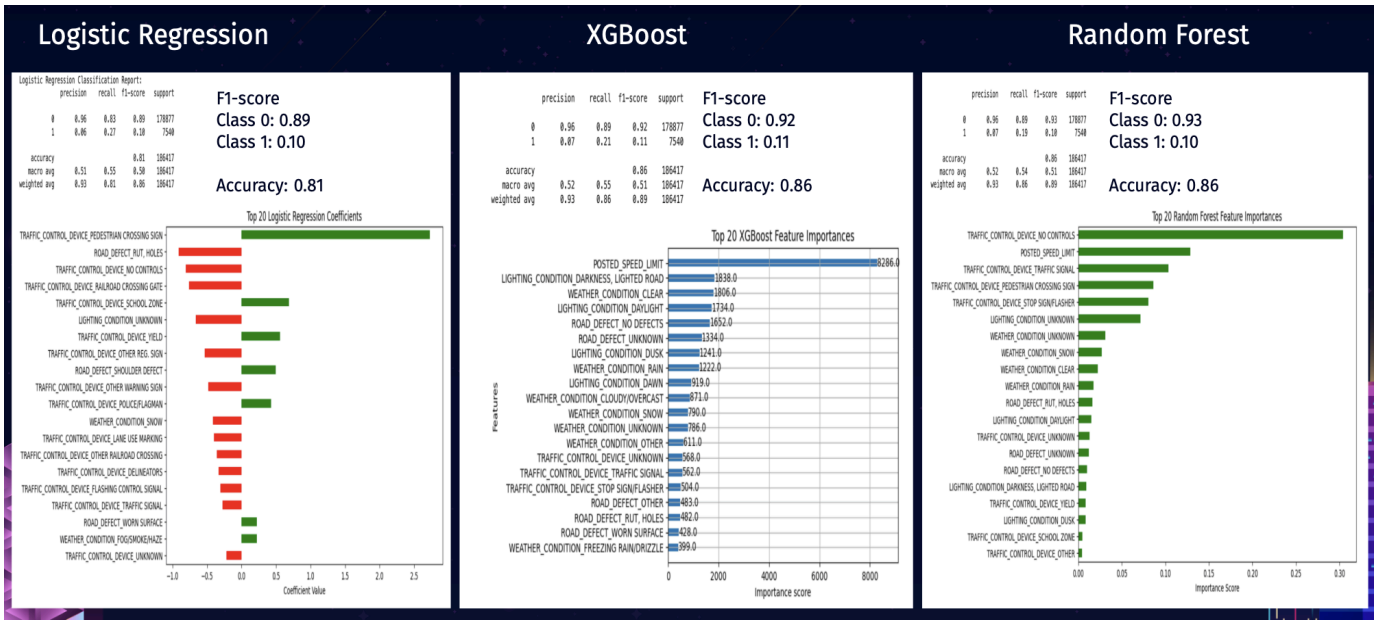
2c: XG Boost

For our XG Boost model, we did the largely same thing as above for random forest and logistic regression; imported a package and fit the model to our dataset based on the training set. In the model it is important to note the class weights we assigned, and how class 1 (pedestrian related) is 18 times that of class 0 (non-pedestrian). We did this for the same reason as listed above in logistic regression and random forest. Similarly, we set the max depth to be 10 to ensure that the model did not overfit the data too heavily by using too many features that were essentially irrelevant. Below is the code snippet for creating and fitting the model for our data.

```
xgb_model = xgb.XGBClassifier(scale_pos_weight=18, n_estimators=200, max_depth=10,
learning_rate=0.05)
xgb_model.fit(X_train, y_train)
```

Results:

After running all these models, we decided to plot the most important features to make it easy to see and interpret which features the models found to be most helpful for learning.



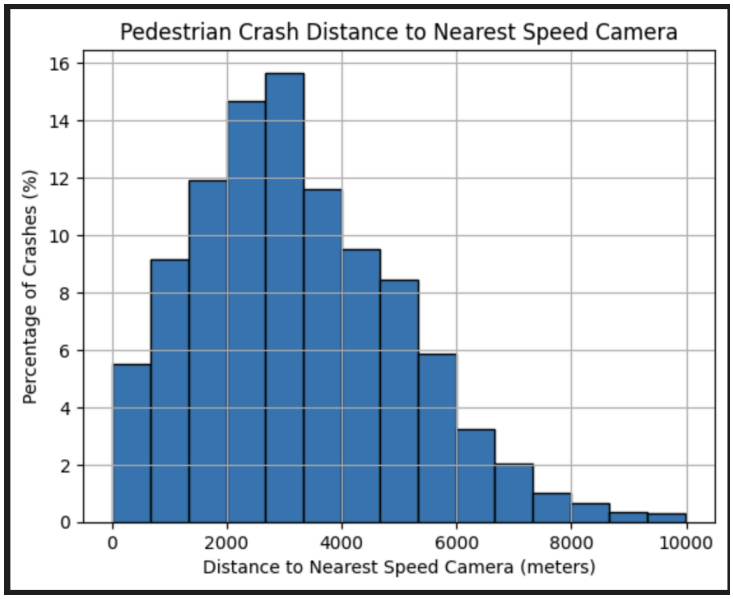
As we can see, all these models struggled to properly predict class 1 (pedestrian related incidents) and tended to just choose class 0 most of the time. This is likely due to the heavy class

imbalance (~4% pedestrian related out of ~1 million data points) as well as label ambiguity. Overall for this ML analysis, we can see that the models did not serve as good predictors of whether a crash would be pedestrian / non-pedestrian. However, that is okay, as the goal of this ML analysis was not to predict crashes, but to find which features were most important in determining whether or not a crash was pedestrian related. We can still accomplish this through analyzing the 3 models and their coefficients / feature importances, as even though the models are not great predictors, they did learn helpful decision boundaries which tell us the features that are most important in determining pedestrian / non pedestrian accidents!

Third Analysis: Analyze Effectiveness of Speed Cameras on Pedestrian Incidents

Initially in our project proposal, we hypothesized that speed cameras would be an effective means to reducing pedestrian related incidents. In this final analysis, we wanted to determine how effective speed cameras actually are in reducing pedestrian related incidents to evaluate them as a potential traffic device / safety measure to employ in the hotspots we found in the first analysis. To do this, we built a KD Tree based on speed camera locations, and for each pedestrian related crash queried the tree, essentially asking it "what is the closest speed camera to this crash?", and visualized these distances using the histogram below.

Results:



From the plot above, the histogram shows us a fairly normal distribution with some skew to the left, indicating that we can not definitively say speed cameras reduce pedestrian related incidents and are the sole solution. From the plot we can say that most incidents are far away from cameras and less are close, but the KD Tree analysis does not show speed cameras as the definitive top option. This analysis was very important for us, as it forced us to pivot away from speed cameras as our main tool to reducing pedestrian incidents and look for other options as well!

Conclusion

Throughout this project, we have gathered lots of insights from our EDA, Visualizations, and ML analyses about traffic crashes across the city of Chicago, with special attention to pedestrian related incidents. We explored different hypotheses to determine important aspects of pedestrian crashes, such as the most impactful weather & lighting conditions, what times and days were most dangerous, and many others. We created an interactive map to allow a user to freely explore pedestrian crashes in the city of Chicago through a simple, easy to navigate interface. We developed different ML models to perform feature extraction to find the most important features for pedestrian accidents, performed clustering analyses to find hotspots across different neighborhoods, and analyzed the effectiveness of speed cameras on pedestrian incidents with ML techniques. Overall, we thoroughly explored the traffic crash dataset for Chicago and provided many meaningful visualizations and inferences that could be very useful in the hands of lobbyists and activists seeking to improve walkability and pedestrian safety in the city of Chicago. Although we did not achieve all we set out to, we believe that with all the work and insights we have amassed, we have just about reached the goal of providing and proposing data driven improvements to pedestrian safety and walkability in Chicago!