Data Science Project Documentation:

Introduction:

Road Traffic Accidents continue to be a significant concern all over the world, resulting in loss of life, property damage, injuries, and economic burden. Road traffic accidents are a global crisis, claiming the lives of 1.3 million people and causing nearly 50 million non-fatal injuries annually. To mitigate these incidents and improve road safety, it is crucial to identify their root causes.

So, traffic crash data analysis plays a pivotal role in understanding the dynamics of road safety, assisting policymakers and law enforcement agencies in implementing effective measures to reduce accidents and their impacts.

Dataset:

Crash Reports – Incidents Data – Maryland, USA.

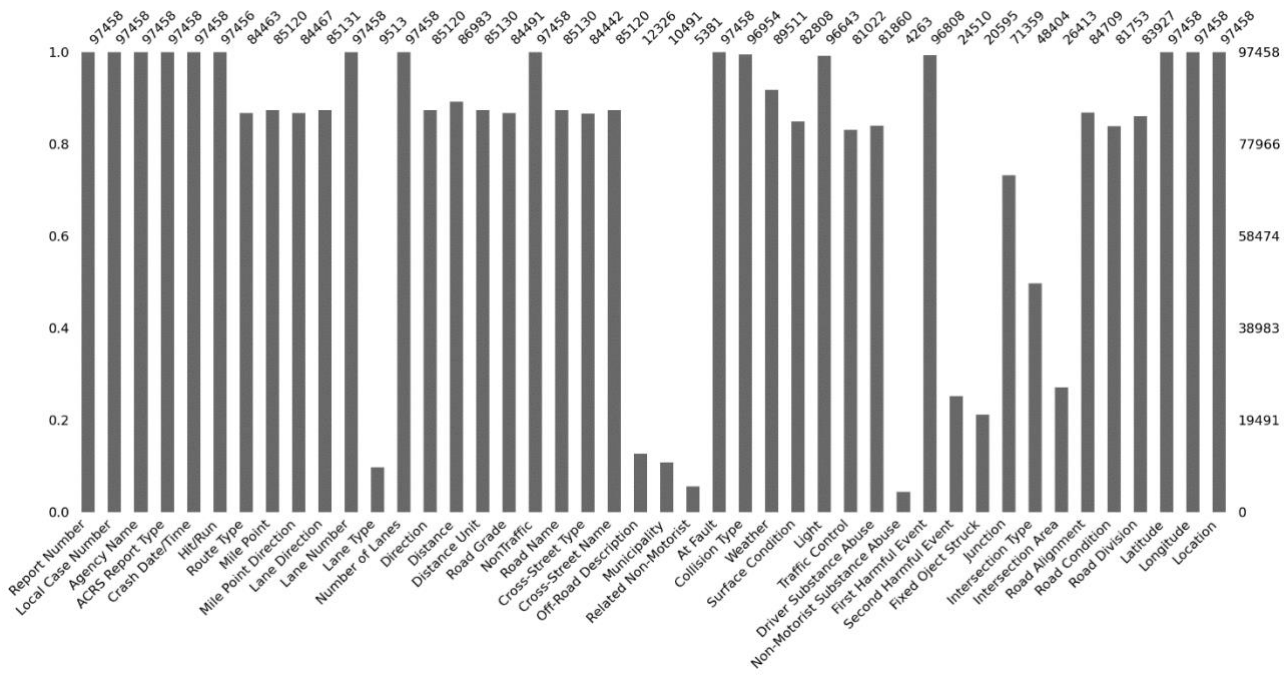
About the data:

This dataset provides general information about each collision and details of all traffic collisions occurring on county and local roadways within Montgomery County, as collected via the Automated Crash Reporting System (ACRS) of the Maryland State Police, and reported by the Montgomery County Police, Gaithersburg Police, Rockville Police, or the Maryland-National Capital Park Police**.**

- 44 Columns of Data with 97,458 Rows. Each row corresponds to a Collision.
- Based on the literature review I've conducted; I've found that the below features are crucial in solving problems related to traffic crashes:
    - ACRS Report Type, Hit/Run, Road Grade, Related Non-Motorist, At Fault, Collision Type, Weather, Surface Condition, Light, Traffic Control, Driver Substance Abuse, Road Condition and Location are the most relevant features in this dataset.
- Many features had a lot of null values and ambiguous categories which were dealt with in the Data Cleaning process.

Data Cleaning:

- Initially, I have sorted the data according to date and time, which helps for time series analysis.
- After checking out all the categorical columns and their unique values, I have found many redundant categories/duplicate categories.
    - Features like 'Agency Name', 'Route Type' and 'Driver Substance Abuse' had many redundant categories. Using .replace() from pandas, I've replaced these redundant categories.
- I've found many features with a lot of null values that can be seen below:



- The data has two types of crashes – On Road and Off Road. Since the number of data points for Off Road crashes are low in number (12,328) and the features related to off-road are only 14 of them. I've decided to drop crashes related to Off-Road.
- Due to this a lot of null values disappeared since they were related to Off-Road crashes while most of the features are related to On-Road crashes.
- Since we removed columns related to off road, I dropped features like 'Off-Road Description', 'Cross-Street Name' and 'Road Name' (Text/NLP features).

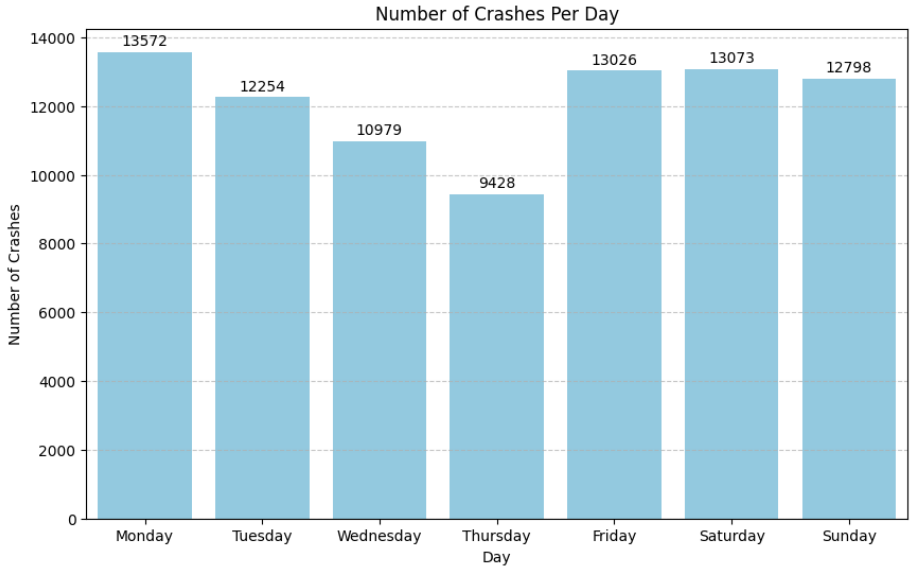
- Categorical:
    - Here, two features 'Related Non-Motorist' and 'Non-Motorist Substance Abuse' have almost 80,000+ null values. Assuming that when there is no 'non-motorist' involved in the crash, these values are null. So instead of leaving it null, I am filling with 'NONE' for those where both the features are null at the same time (new category when there is no non motorist involved).
    - Similarly, I've filled inter-related features like 'Intersection Area' and 'Intersection Type' that are both NaN with 'None', and filled the rest with 'UNKNOWN'.
    - The remaining null valued columns were filled with UNKNOWN.
- Numerical:
    - Features like 'Distance' and 'Mile Point' having null values were filled using **Random Sampling** – a probability-based technique.
- Now there are 0 null values in the entire dataset.

Exploratory Data Analysis (EDA):

****Only important and relevant plots/graphs were documented here. The rest of the plots that I felt irrelevant are in jupyter notebook****

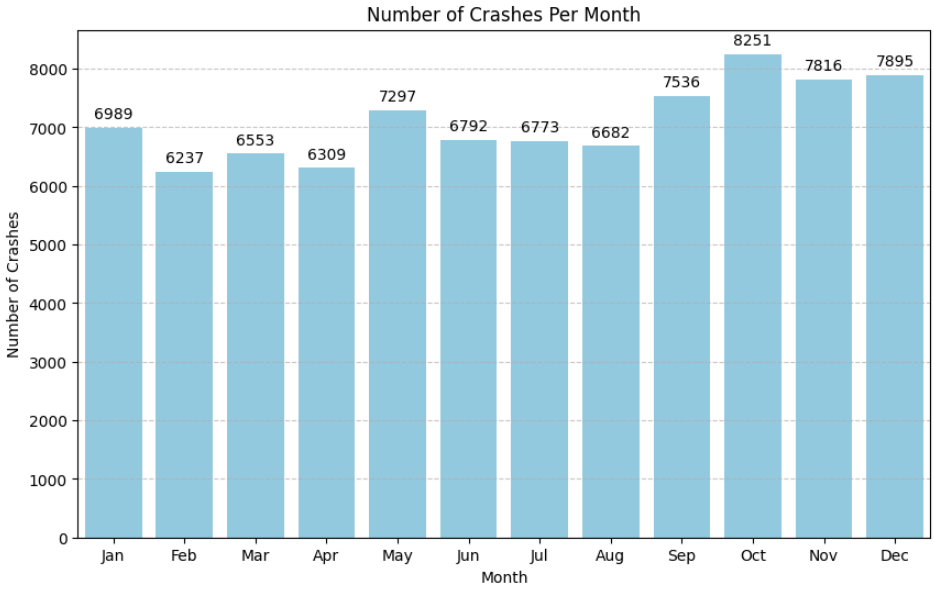
Time Series Analysis:

1. Number of Crashes per day:



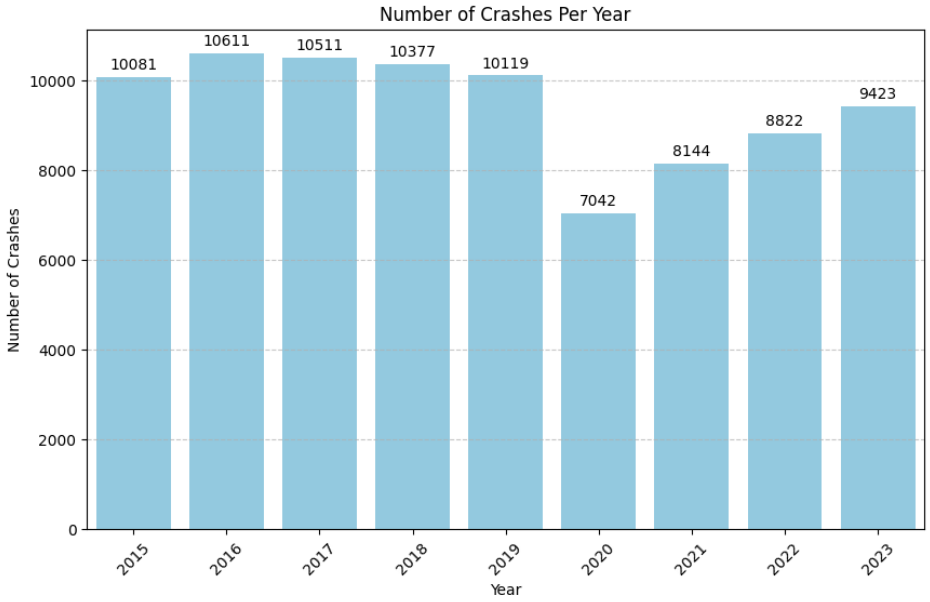
Observation: Less number of crashes on Thursdays.

2. Number of crashes per month:



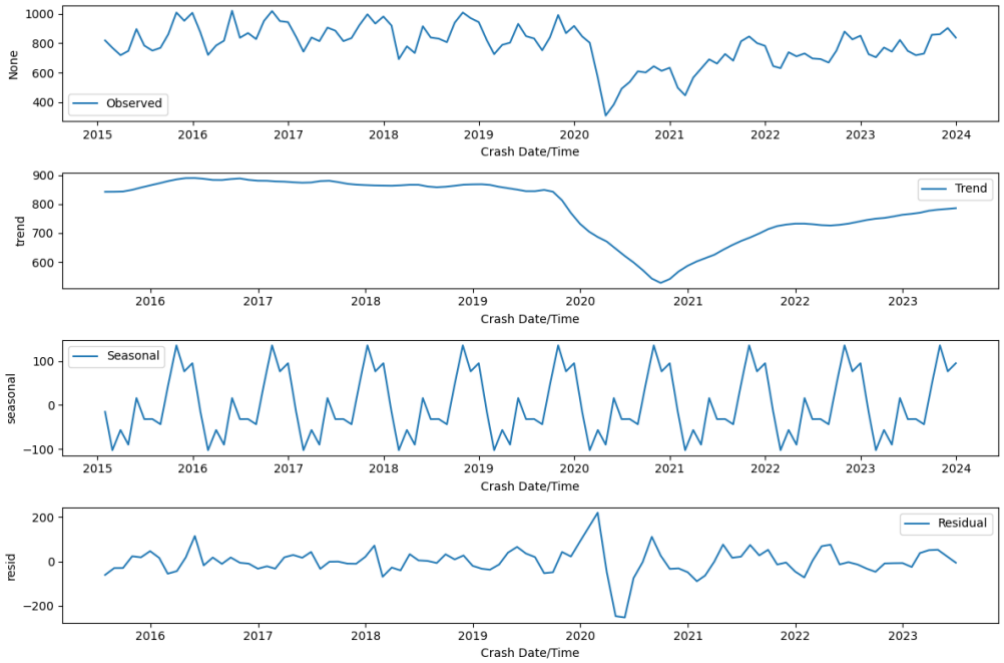
Observation: High number of crashes during October and December probably due to the Holiday season (Halloween and Christmas).

3. Number of Crashes Per Year:



Observation: Year 2020 and 2021 had the least number of crashes probably due to COVID-19.

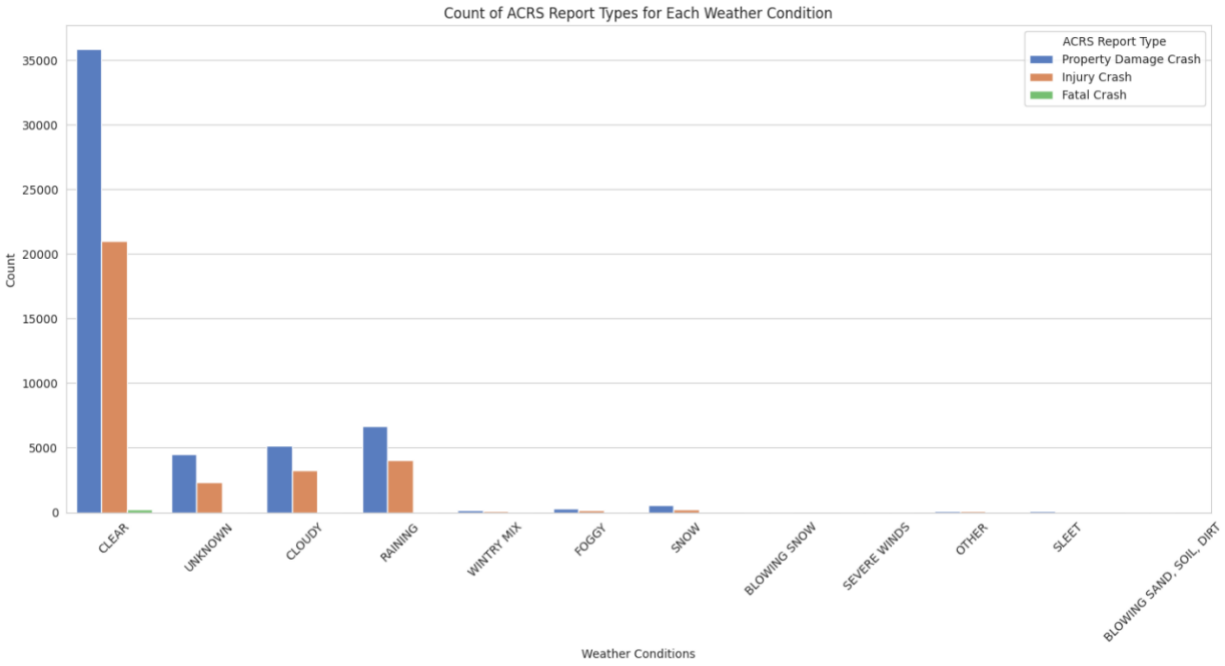
4. Seasonal Decomposition for Monthly Crashes:



Observation: There is a pattern in the seasonal component of the number of crashes in a monthly basis. So, the same pattern of crashes is being followed monthly, every year.

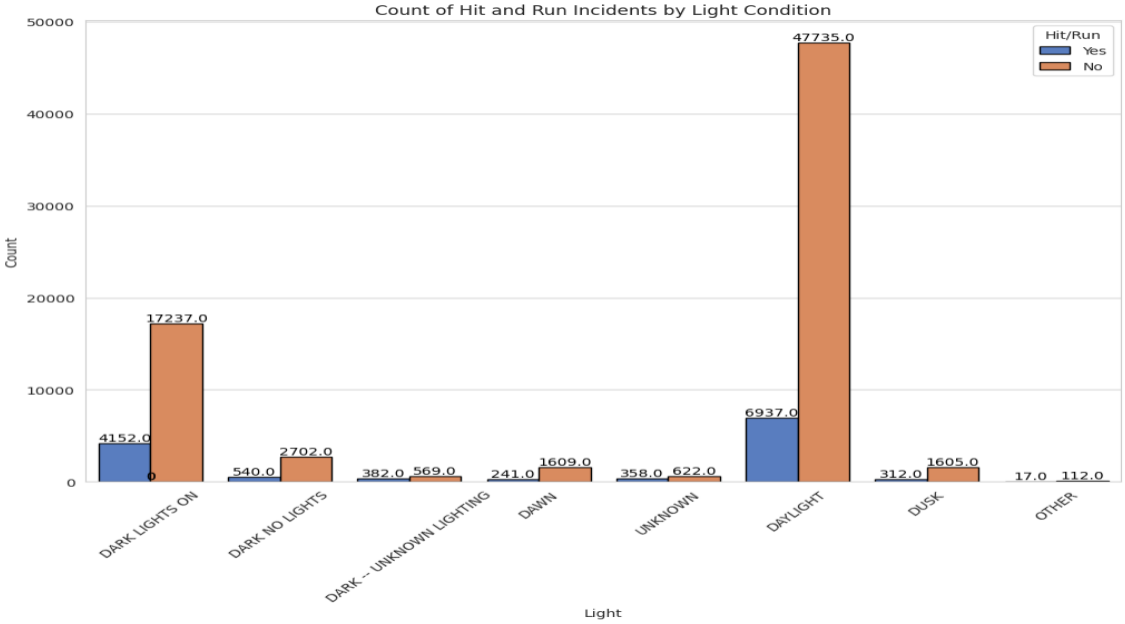
Feature Analysis:

1. Count of ACRS Report Types for Each Weather Condition:



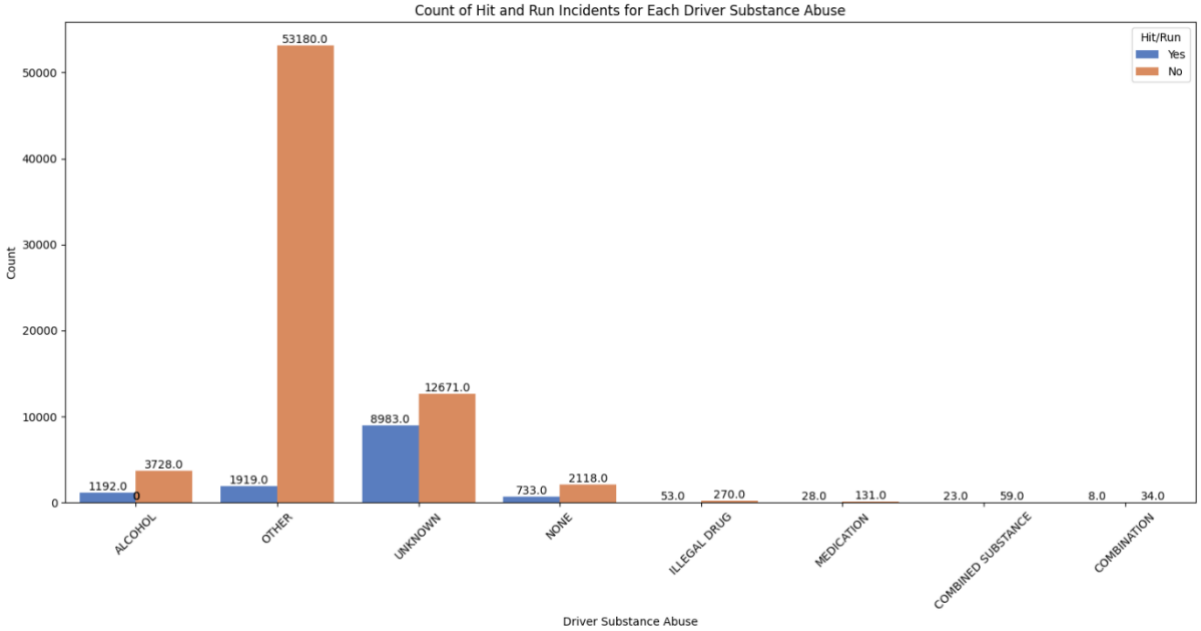
Observation: Most of the accidents that are property damage crash and injury crashes are during Clear weather

2. Count of Hit and Run incidents by Light conditions:



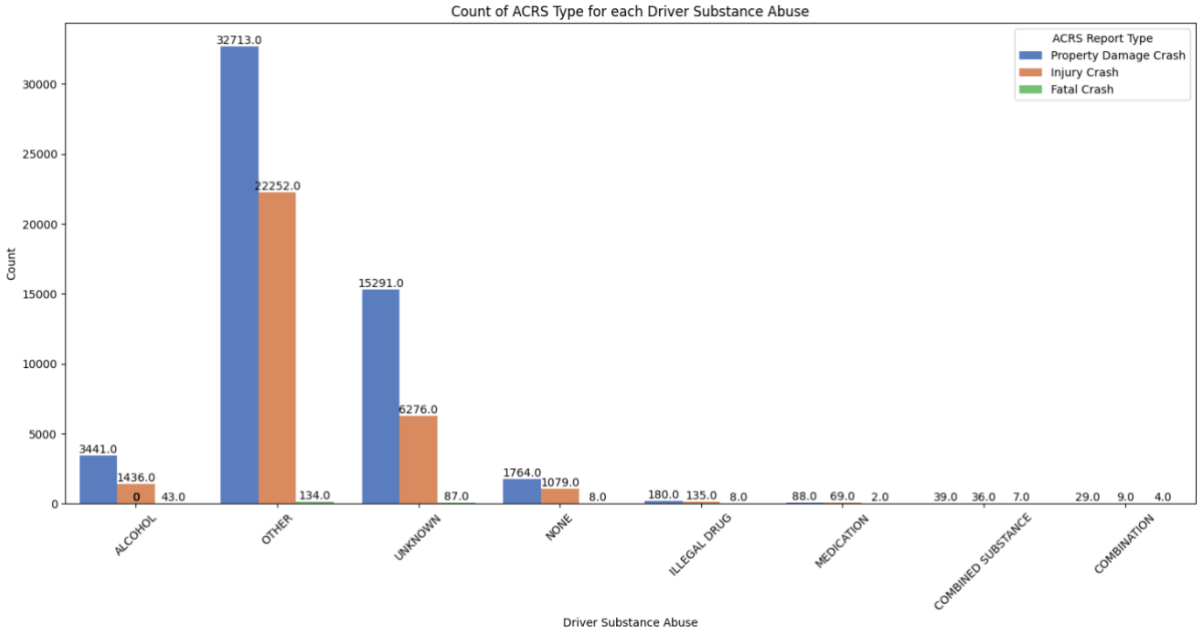
Observation: More Hit and Run Cases during dark with lights on and even during daylight.

3. Count of Hit and Run with Driver Substance Abuse:



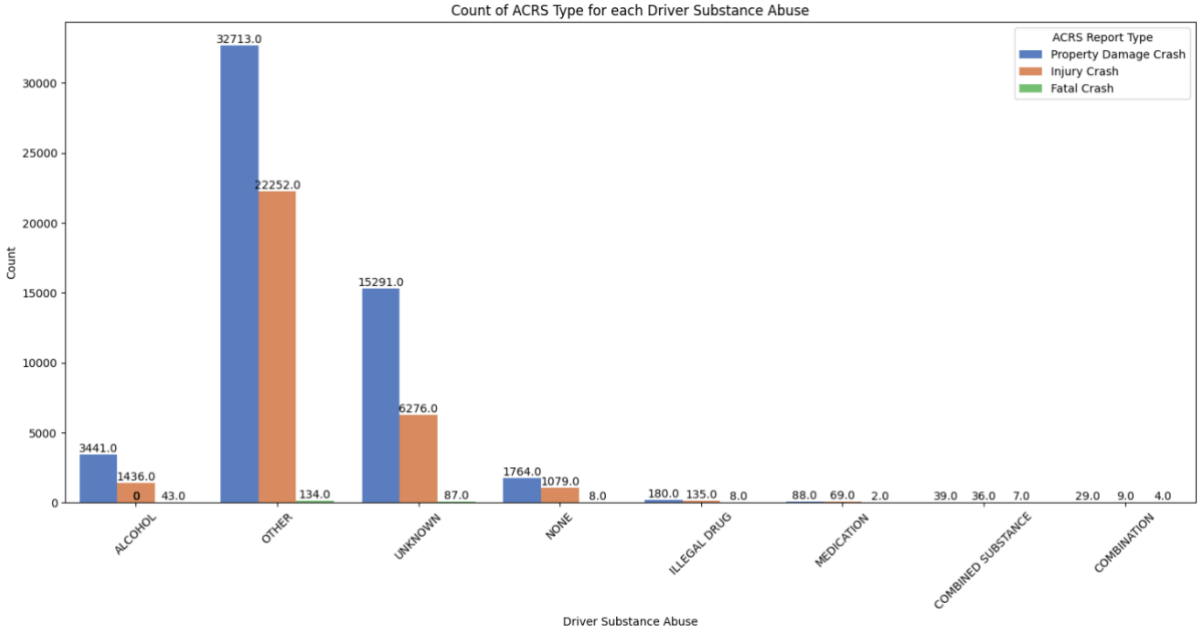
Observation: Many of the Hit and Run cases were due to Alcohol, Unknown substance abuse and Other substance abuse.

4. Count of ACRS Report Type for each Driver Substance Abuse:



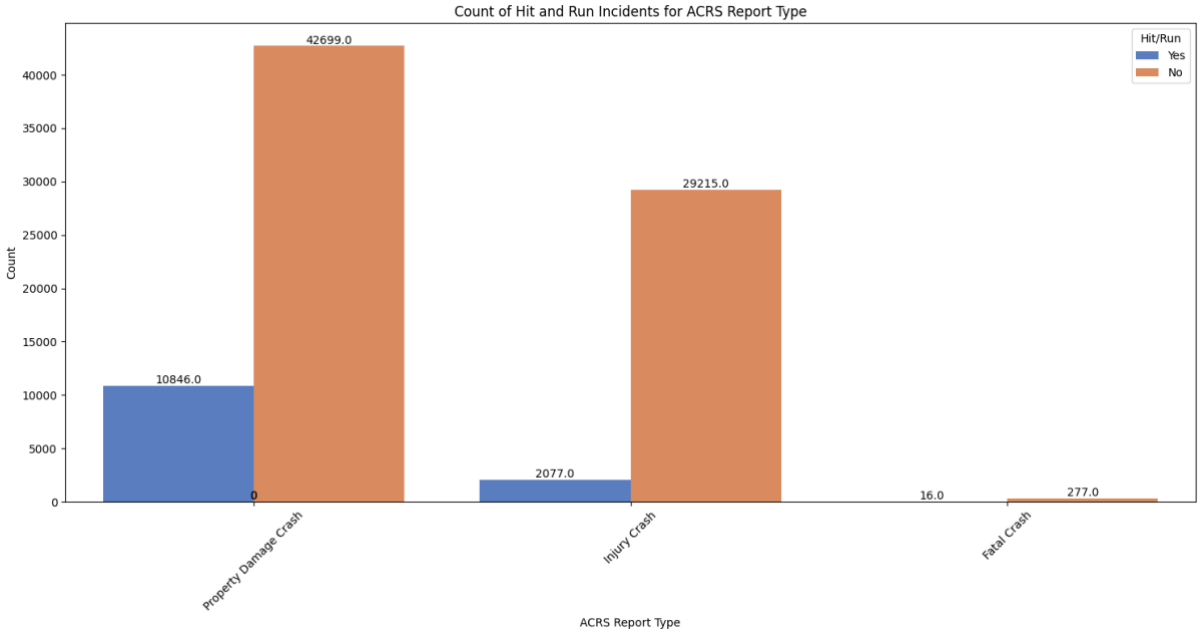
Observation: More injury crashes with Other and Unknown Substance Abuse. Majority of the Fatal Crashes are due to Illegal Drug and Other Substance Abuse.

5. Collision Type and Crash Severity:



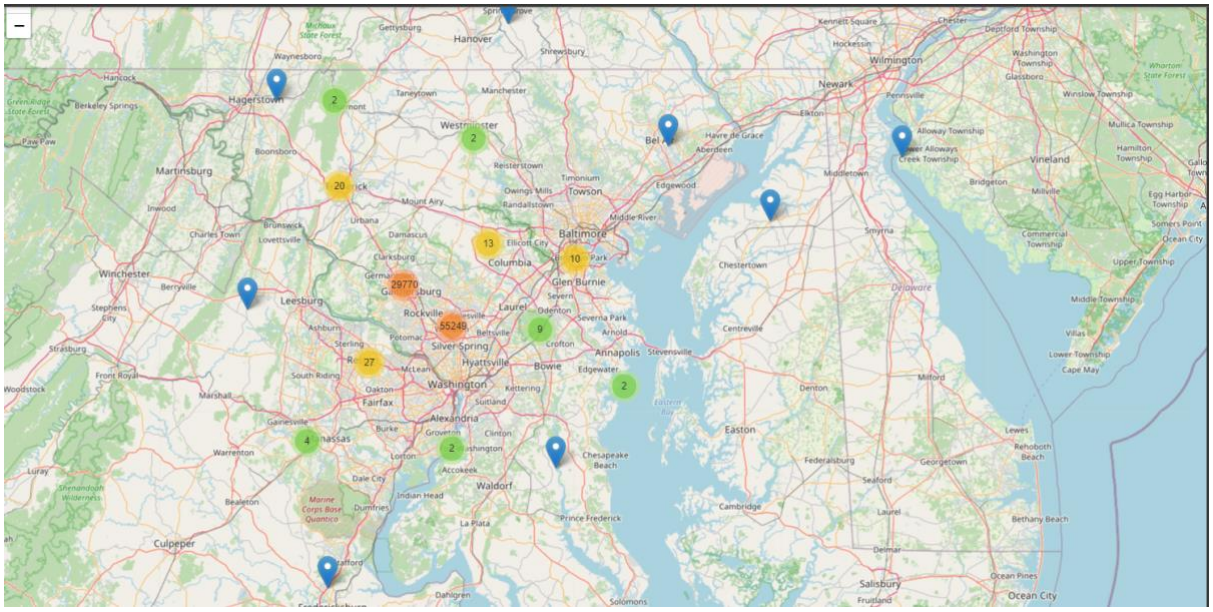
Observation: Many Fatal Crashes are due to Opposite Direction Sideswipe (90-degree crashes) and Angle meets Left Turn Crashes.

6. ACRS and Hit/Run:



Observation: Most of the accidents that are Hit and Run were during Property Damage Crashes.

Spatial Analysis:



The above map provides a various hotspots of crashes along with crash frequency in the state. The interactive map in the python notebook will give a better idea to determine which location/sub-location has more frequency of crashes.

Correlational Analysis:

1. Chi-Square Test for Categorical Variables:
    a. According to the test, if p-value between 2 variables is less than 0.05, it means the features are correlated. The closer it is to 0.05, the more correlated they are and vice versa.
    b. From the table, it can be inferred that there is a very little to no correlation among the features since the p-values are very small (in powers of 10^-2 to 10^-200s)

2. Cramer's V Correlation for Categorical Variables:
    a. This is another test to check the correlation among all categorical variables in the data.
    b. From the matrix, it can be seen that the highest correlation of 0.58 is between 'Mile Point Direction' and 'Direction'.
    c. This did not give good enough understanding

3. Theil's U Correlation for Categorical Variables:
    a. From this correlation matrix, features like 'Related Non-Motorist' and 'Non-Motorist Driver Substance Abuse' had high correlation of 0.82;
    b. Also, 'Related Non-Motorist' and 'First Harmful Event' had high correlation of 0.78.

Machine Learning Models:

Before starting the model building, I have converted all categorical features into numerical using Label Encoder.

Problem 1: Predicting the severity of collisions

- Target Variable: ACRS Report Type (3 Categories)
- Input features: Weather, Road Condition, At Fault, Driver Substance Abuse, Road Alignment, Surface Condition, Non-Motorist Substance Abuse, Light and Collision Type
- Feature Selection: Recursive Feature Elimination with Decision Trees and Random Forests gave similar features.
- Models:
    - Logistic Regression:
        - Train Accuracy: 0.6290
        - Test Accuracy: 0.6281
        - F1-Score: 0.63
    - Decision Trees:
        - Train Accuracy: 0.8156
        - Test Accuracy: 0.6098
        - F1-Score: 0.61
    - Random Forests:
        - Train Accuracy: 0.9996
        - Test Accuracy: 0.6608
        - F1-Score: 0.66
    - XGBoost:
        - Train Accuracy: 0.9363
        - Test Accuracy: 0.6286
        - F1-Score: 0.63
- Best Model – Random Forests with highest F1-Score and Testing accuracy.

Problem 2: Hit and Run Prediction: Predicting a vehicle involved in collision will flee or not

- Target Variable: Hit/Run (2 Categories)
- Input features: Location, Traffic Control, Weather, Road Condition, At Fault, Driver Substance Abuse, Road Alignment, Surface Condition, Non-Motorist Substance Abuse, Light and Collision Type
- Models:
    - Logistic Regression:
        - Train Accuracy: 0.8480
        - Test Accuracy: 0.8480
        - F1-Score: 0.85
    - Decision Trees:
        - Train Accuracy: 1.0
        - Test Accuracy: 0.8199
        - F1-Score: 0.82
    - Random Forests:
        - Train Accuracy: 0.9998

        - Test Accuracy: 0.8628
        - F1-Score: 0.86
    - XGBoost:
        - Train Accuracy: 0.9005
        - Test Accuracy: 0.8746
        - F1-Score: 0.87
- Best Model - XGBoost with highest F1-Score and Testing accuracy without any overfitting.

Problem 3: Predicting parties at Fault: Determining which parties are at fault in a collision.

- Target Variable: At Fault
- Input features: Location, Traffic Control, Weather, Road Condition, At Fault, Driver Substance Abuse, Road Alignment, Surface Condition, Non-Motorist Substance Abuse, Light and Collision Type
- Due to high imbalance in the categories of Y, I'm using weighted loss function to balance the class weights.
- Models:
    - Logistic Regression:
        - Train Accuracy: 0.4271
        - Test Accuracy: 0.4290
        - F1-Score: 0.46
    - Decision Trees:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.8911
        - F1-Score: 0.89
    - Random Forests:
        - Train Accuracy: 0.9998
        - Test Accuracy: 0.9294
        - F1-Score: 0.93
    - XGBoost:
        - Train Accuracy: 0.9465
        - Test Accuracy: 0.9312
        - F1-Score: 0.93
- Best Model – XGBoost with highest F1-Score and Testing accuracy.

Problem 4: Driver Substance Abuse detection

- Target Variable: Driver Substance Abuse
- Input features: ACRS Report Type, Hit/Run, Road Grade, Municipality, Related Non-Motorist, At Fault, Collision Type, Weather, Surface Condition, Light, Traffic Control, First Harmful Event, Intersection Type, Road Condition, Road Alignment, Latitude, Longitude
- Using weighted loss function to mitigate class imbalance.
- Models:
    - Logistic Regression:
        - Train Accuracy: 0.3283
        - Test Accuracy: 0.3296
        - F1-Score: 0.33
    - Decision Trees:

        - Train Accuracy: 0.9999
        - Test Accuracy: 0.5825
        - F1-Score: 0.58
    - Random Forests:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.7176
        - F1-Score: 0.72
    - XGBoost:
        - Train Accuracy: 0.7610
        - Test Accuracy: 0.7340
        - F1-Score: 0.93
- Best Model –XGBoos with highest F1-Score, testing accuracy and not much overfitting.

Problem 5: Collision Type Detection:

- Target Variable: Collision Type
- Input features: ACRS Report Type, Hit/Run, Road Grade, At Fault, Weather, Surface Condition, Light, Traffic Control, First Harmful Event, Intersection Type, Road Condition, Road Alignment, Latitude, Longitude
- Using weighted loss function to mitigate class imbalance.
- Models:
    - Logistic Regression:
        - Train Accuracy: 0.1021
        - Test Accuracy: 0.1019
        - F1-Score: 0.10
    - Decision Trees:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.2854
        - F1-Score: 0.29
    - Random Forests:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.3740
        - F1-Score: 0.37
    - XGBoost:
        - Train Accuracy: 0.5308
        - Test Accuracy: 0.4277
        - F1-Score: 0.43
- Best Model – None. Due to extreme class imbalance, the models are getting overfitted. Hyperparameter tuning or synthetic data sampling methods might help in increasing the accuracies and f1-score for this problem statement.

Problem 6: Junction Prediction: Predicting which type of Junctions crashes occur:

- Target Variable: Junction
- Input features: 'ACRS Report Type', 'Hit/Run', 'At Fault', 'Collision Type', 'Weather', 'Surface Condition', 'Light', 'Traffic Control', 'Road Condition', 'Driver Substance Abuse', 'Intersection Type', 'Road Alignment', 'Road Grade', 'Road Division', 'Latitude', 'Longitude'.
- Using weighted loss function to mitigate class imbalance.

- Models:
    - Logistic Regression:
        - Train Accuracy: 0.2295
        - Test Accuracy: 0.2291
        - F1-Score: 0.23
    - Decision Trees:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.5494
        - F1-Score: 0.55
    - Random Forests:
        - Train Accuracy: 0.9999
        - Test Accuracy: 0.6533
        - F1-Score: 0.65
    - XGBoost:
        - Train Accuracy: 0.7503
        - Test Accuracy: 0.6671
        - F1-Score: 0.67
- Best Model – XGBoost, but Due to extreme class imbalance, the models are getting overfitted. Hyperparameter tuning or synthetic data sampling methods might help in increasing the accuracies and f1-score for this problem statement.

Results:

- For problem statements 1, 2 and 3 the machine learning models predict good enough without much overfitting.
- For the other problem statements, due to the lack of equal number of samples in each category of target variable, the models are overfitting by predicting the same class for most of the test data points, giving it high training accuracies and low testing accuracy.

Future Works:

- SMOTE – Using Synthetic sampling technique to balance the imbalanced classes for each problem.
- Hyperparameter Tuning for every model in every problem statement might result in the most optimal model with decent accuracies.
- Using other feature elimination techniques apart from RFE to get important features.
- Using Feature Reduction Techniques like PCA/LDA/Autoencoders to reduce number of features might give better results.
- Using ANNs to solve different problem statements as the data might have non-linear relationships.
- Adding external features like weather data (temperature, windspeed, etc.) using Historical Weather API might benefit in solving few problem statements.
- Time Series Forecasting of number of crashes on a monthly basis or daily basis.

Conclusion:

In this Data Science project, we delved into the road traffic collision data from Maryland, USA, aiming to gain insights into the factors influencing traffic accidents and improve road safety measures. Through the exploration and modeling of various features, I have found several key findings. Overall,

this project underscores the importance of data-driven approaches in addressing road safety challenges. Leveraging machine learning and data analytics can lead to the development of effective strategies for reducing accidents, saving lives, and safer roads.