CAPSTONE PROJECT – CRASH REPORTING ANALYSIS

INFO 6105: Data Science Engineering Management

College of Engineering, Northeastern University

April 22, 2024

Submitted by,

Swanand Chandrakant Tanavade
NUID: 002774342

INTRODUCTION

The selected dataset for analysis encompasses detailed information on vehicular crashes in Montgomery County, Maryland, and encompasses driver-related information for each incident. Comprising 172,105 entries and 43 distinct data points, the dataset collates reports from several agencies, such as the Montgomery County Police and other local law enforcement agencies, sourcing data from the Maryland State Police's Automated Crash Reporting System. Accessible through the U.S. Government's DATA.GOV website, this repository is designed to simplify the collection, study, and dissemination of vehicular crash data for purposes ranging from law enforcement to traffic safety research.

This dataset contains a breadth of information, including but not limited to, demographics of drivers, details on driving behavior, traffic conditions, and specifics of the involved vehicles. Noteworthy data points encompass "Report Number," "Local Case Number," "Agency Name," "Crash Date/Time," "Route Type," "Road Name," "Vehicle Year," "Vehicle Make," "Vehicle Model," and precise geolocation details, among others. With a predominant focus on categorical attributes, the dataset also provides contextual information such as crash dates and locations, to enhance understanding of the crash circumstances.

The core analytical question posits whether the severity of injuries sustained in crashes can be predicted through classification models in machine learning. By examining the interplay between various factors and the severity of injuries, it is possible to construct a model capable of making such predictions. Recognizing patterns and interdependencies among the variables can enable businesses and public entities to anticipate high-risk situations, better allocate resources, and develop strategies to reduce the severity of crash-related injuries. Such a model could significantly bolster decision-making processes and promote the implementation of specific safety measures to diminish the effects of vehicular crashes.

EXPLORATORY DATA ANALYSIS

Packages used:

Pandas: It offers the data structures and functions required to effectively work with structured data. Series and DataFrame, which are based on NumPy arrays, are the two main data structures in Pandas.

Plotly: It enables the creation of engaging data visualizations and interactive charts. Line plots, scatter plots, bar plots, pie charts, and other chart formats are among the many that are offered.

Sklearn: A well-known open-source Python library for machine learning is Scikit-learn. For numerous machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and data preprocessing, it offers a complete collection of tools and functionalities. The following packages were used under sklearn.

- sklearn.metrics: offers a variety of indicators for assessing the effectiveness of machine learning models.
- sklearn.ensemble: a collection of ensemble-based machine learning methods is provided.
- sklearn.preprocessing: offers a number of tools for feature engineering and data preprocessing.
- sklearn.model_selection: provides a collection of instruments for model evaluation and choice.

Importing the dataset:

To import the dataset, a function called “pd.read_csv” was used which was imported from a .py file. A glimpse of the dataset is shown in Figure 1.

	Report Number	Local Case Number	Agency Name	ACRS Report Type	Crash Date/Time	Route Type	Road Name	Cross-Street Type	Cross-Street Name	Off-Road Description	...	Speed Limit	Driverless Vehicle	Parked Vehicle	Vehicle Year	Vehicle Make	Vehicle Model	Equipment Problems	Latitude	Longitude	Location
0	MCP3040003N	190026050	Montgomery County Police	Property Damage Crash	05/31/2019 03:00:00 PM	NaN	NaN	NaN	NaN	PARKING LOT OF 3215 SPARTAN RD	...	15	No	No	2004	HONDA	TK	UNKNOWN	39.150044	-77.063089	(39.15004368, -77.06308884)
1	EJ7885038	230034791	Gaithersburg Police Depar	Property Damage Crash	07/21/2023 05:59:00 PM	Maryland (State)	FREDERICK RD	Unknown	WATKINS MILL RD	NaN	...	40	No	No	2011	GMC	TK	NO MISUSE	39.159264	-77.219025	(39.1592635, -77.21902483)
2	MCP2009002G	230034583	Montgomery County Police	Property Damage Crash	07/20/2023 03:10:00 PM	Maryland (State)	GEORGIA AVE	Maryland (State)	NORBECK RD	NaN	...	35	No	No	2019	FORD	F150	NO MISUSE	39.109535	-77.075806	(39.10953506, -77.07580619)
3	MCP3201004C	230035036	Montgomery County Police	Property Damage Crash	07/23/2023 12:10:00 PM	County	CRYSTAL ROCK DR	County	WATERS LANDING DR	NaN	...	40	No	No	2016	KIA	SW	NO MISUSE	39.190149	-77.266766	(39.19014917, -77.26676583)
4	MCP23290028	230035152	Montgomery County Police	Property Damage Crash	07/24/2023 06:10:00 AM	County	MONTGOMERY VILLAGE AVE	County	CENTERWAY RD	NaN	...	35	No	No	2016	TOYT	TK	NO MISUSE	39.172558	-77.203745	(39.17255801, -77.20374546)

Figure 1

The information about the dataset is illustrated in Figure 2. Each column represents a different attribute of the dataset, and the non-null count indicates the number of non-missing values present in each column. The dataset contains a mix of data types, including object (representing strings), int64 (representing integers), and float64 (representing floating-point numbers). Some columns have a significant number of missing values, as indicated by the difference between the total number of entries (172,105) and the non-null count for each column.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172105 entries, 0 to 172104
Data columns (total 43 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Report Number                   172105 non-null  object
 1   Local Case Number               172105 non-null  object
 2   Agency Name                     172105 non-null  object
 3   ACRS Report Type                172105 non-null  object
 4   Crash Date/Time                 172105 non-null  object
 5   Route Type                      155132 non-null  object
 6   Road Name                       156168 non-null  object
 7   Cross-Street Type               155099 non-null  object
 8   Cross-Street Name               156154 non-null  object
 9   Off-Road Description            15935 non-null   object
 10  Municipality                    19126 non-null   object
 11  Related Non-Motorist            5463 non-null    object
 12  Collision Type                  171520 non-null  object
 13  Weather                         158751 non-null  object
 14  Surface Condition               151987 non-null  object
 15  Light                           170660 non-null  object
 16  Traffic Control                 146636 non-null  object
 17  Driver Substance Abuse          140781 non-null  object
 18  Non-Motorist Substance Abuse    4317 non-null    object
 19  Person ID                       172105 non-null  object
 20  Driver At Fault                 172105 non-null  object
 21  Injury Severity                 172105 non-null  object
 22  Circumstance                    31359 non-null   object
 23  Driver Distracted By            172105 non-null  object
 24  Drivers License State           162155 non-null  object
 25  Vehicle ID                      172105 non-null  object
 26  Vehicle Damage Extent           171789 non-null  object
 27  Vehicle First Impact Location   171949 non-null  object
 28  Vehicle Second Impact Location  171849 non-null  object
 29  Vehicle Body Type               169456 non-null  object
 30  Vehicle Movement                171719 non-null  object
 31  Vehicle Continuing Dir          169416 non-null  object
 32  Vehicle Going Dir               169416 non-null  object
 33  Speed Limit                     172105 non-null  int64
 34  Driverless Vehicle              172105 non-null  object
 35  Parked Vehicle                  172105 non-null  object
 36  Vehicle Year                    172105 non-null  int64
 37  Vehicle Make                    172081 non-null  object
 38  Vehicle Model                   172039 non-null  object
 39  Equipment Problems              137964 non-null  object
 40  Latitude                        172105 non-null  float64
 41  Longitude                       172105 non-null  float64
 42  Location                        172105 non-null  object
dtypes: float64(2), int64(2), object(39)
memory usage: 56.5+ MB
```

Figure 2

Descriptive Statistics:

As the dataset contains more numerical values, the descriptive statistics is used to examine only the numerical values as shown in Figure 3. The "Report Number" and "Local Case Number" attributes have 90,958 and 99,608 unique values, respectively, suggesting that there are some

duplicate entries or multiple reports for some cases.

The most frequent occurrence in the "Agency Name" attribute is "Montgomery County Police," and in the "ACRS Report Type" attribute, it is "Property Damage Crash." For numerical attributes like "Speed Limit," "Vehicle Year," "Latitude," and "Longitude," the mean and standard deviation values are provided, indicating the average and spread of the data, respectively.

	Report Number	Local Case Number	Agency Name	ACRS Report Type	Crash Date/Time	Route Type	Road Name	Cross-Street Type	Cross-Street Name	Off-Road Description	...	Speed Limit	Driverless Vehicle	Parked Vehicle	Vehicle Year	Vehicle Make	Vehicle Model	Equipment Problems	Latitude	Longitude	Location
count	172105	172105	172105	172105	172105	155132	156168	155099	156154	15935	...	172105.000000	172105	172105	172105.000000	172081	172039	137964	172105.000000	172105.000000	172105
unique	96854	108781	10	3	94499	10	3803	10	6697	11302	...	NaN	2	2	NaN	1877	6691	10	NaN	NaN	96009
top	MCP229800RC	200022715	Montgomery County Police	Property Damage Crash	12/10/2018 06:10:00 PM	Maryland (State)	GEORGIA AVE	County	GEORGIA AVE	PARKING LOT	...	NaN	No	No	NaN	TOYOTA	4S	NO MISUSE	NaN	NaN	(38.953, -77.338)
freq	10	10	138363	109452	11	77074	10861	85535	2106	132	...	NaN	171367	169416	NaN	23171	13748	123464	NaN	NaN	45
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	32.549752	NaN	NaN	1966.306946	NaN	NaN	NaN	39.083119	-77.112343	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	11.059987	NaN	NaN	340.615468	NaN	NaN	NaN	0.072032	0.098571	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.000000	NaN	NaN	0.000000	NaN	NaN	NaN	37.720000	-79.486000	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	25.000000	NaN	NaN	2006.000000	NaN	NaN	NaN	39.024417	-77.189327	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	35.000000	NaN	NaN	2011.000000	NaN	NaN	NaN	39.074933	-77.105412	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	40.000000	NaN	NaN	2015.000000	NaN	NaN	NaN	39.139742	-77.039592	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	75.000000	NaN	NaN	9999.000000	NaN	NaN	NaN	39.990414	-75.527708	NaN

11 rows × 43 columns

Figure 3

Visualizations:

Figure 4 showcases the top 10 vehicle makes that have contributed to the highest number of accidents. The data reveals that TOYOTA stands out as the vehicle make with the highest involvement in accidents, accounting for over 20,000 incidents. Following closely, HONDA and FORD rank as the second and third highest, respectively, with accident numbers surpassing 15,000. The remaining vehicle makes in the top 10 contributed to less than 10,000 accidents each.

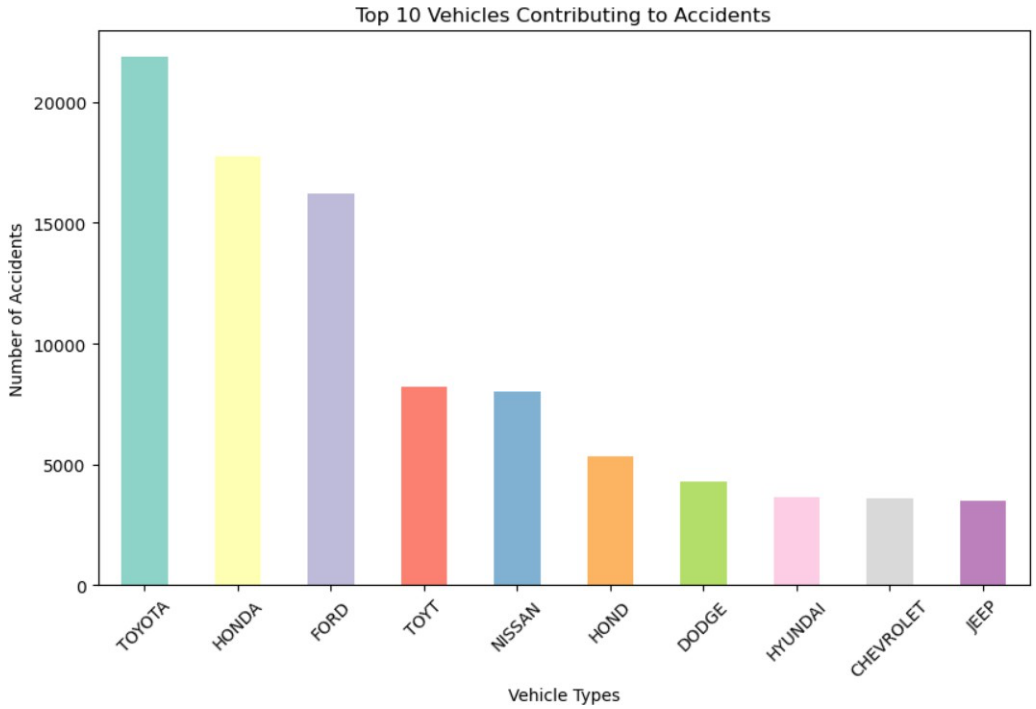

Figure 4

Figure 5 presents a pie chart depicting the types of car crashes recorded in the Automated Crash Reporting System. The chart highlights three distinct categories: crashes resulting in property damage, crashes resulting in injury, and crashes resulting in fatality.

The analysis of the data reveals that a significant majority of the recorded crashes, approximately 63.6%, only caused property damage. This indicates that the majority of accidents had no reported injuries or fatalities, but rather resulted in damage to vehicles or other property. Around 36.2% of the crashes in the dataset resulted in injuries. This suggests a significant portion of accidents had varying degrees of harm to individuals involved, ranging from minor injuries to more severe ones.

Moreover, a small proportion of crashes, constituting only 0.2%, resulted in fatalities. While this percentage is relatively low, it is crucial to acknowledge the gravity of these incidents, as even a small number of fatalities represents significant loss and impact on individuals, families, and communities. By visualizing the distribution of car crash types in this pie chart, it becomes apparent that the majority of accidents primarily involve property damage, with a smaller but still notable portion resulting in injury.

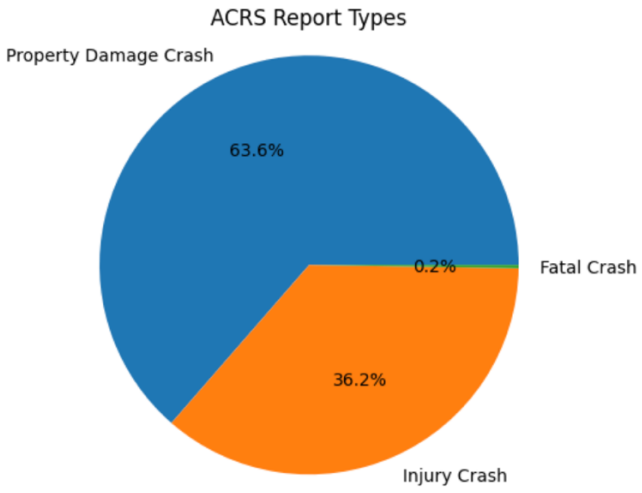

Figure 5

Figure 6 presents a stacked bar graph that depicts various levels of injury severity in relation to driver culpability in vehicle crashes. The categories for injury severity range from fatal to no injury, and from potential to minor and major injuries. The data suggests a predominant trend of driver fault in most accidents. Interestingly, the majority of these incidents report no significant injuries observable at the scene.

In instances where the driver is deemed responsible, the chart shows a comparative decrease in both possible injuries and confirmed minor injuries, hinting that driver-caused accidents might result in a lesser rate of these types of injuries. Conversely, the graph shows a marginal increase in severe injuries for accidents with driver fault, indicating a modest escalation in the risk of serious harm in such situations.

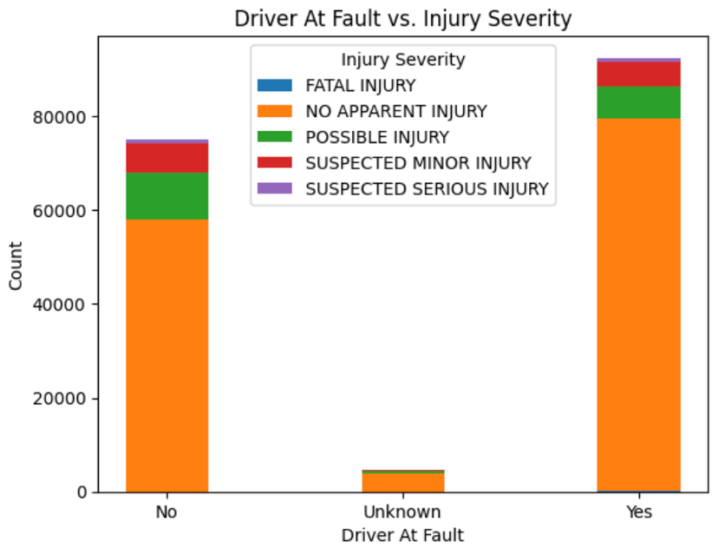

Figure 6

From the scatter plot given in Figure 7, it is evident that the highest number of accidents occurred during daylight with a clear sky, accounting for more than 70,000 accidents. This suggests that accidents are more likely to happen under favorable weather conditions with good visibility. Despite the favorable light conditions, a significant number of accidents still occurred. This highlights the fact that accidents can happen even in clear weather conditions, indicating that factors other than weather, such as driver behavior or road conditions, may also contribute to accidents.

The next most frequent accidents occurred when the weather was cloudy and raining. These weather conditions, particularly in combination with rain, may reduce visibility and affect road conditions, potentially leading to an increased likelihood of accidents.
Furthermore, across all three weather conditions mentioned, the second-highest number of accidents occurred when the dark lights were on. This observation suggests that accidents are more prevalent during darker periods, even when combined with different weather conditions.

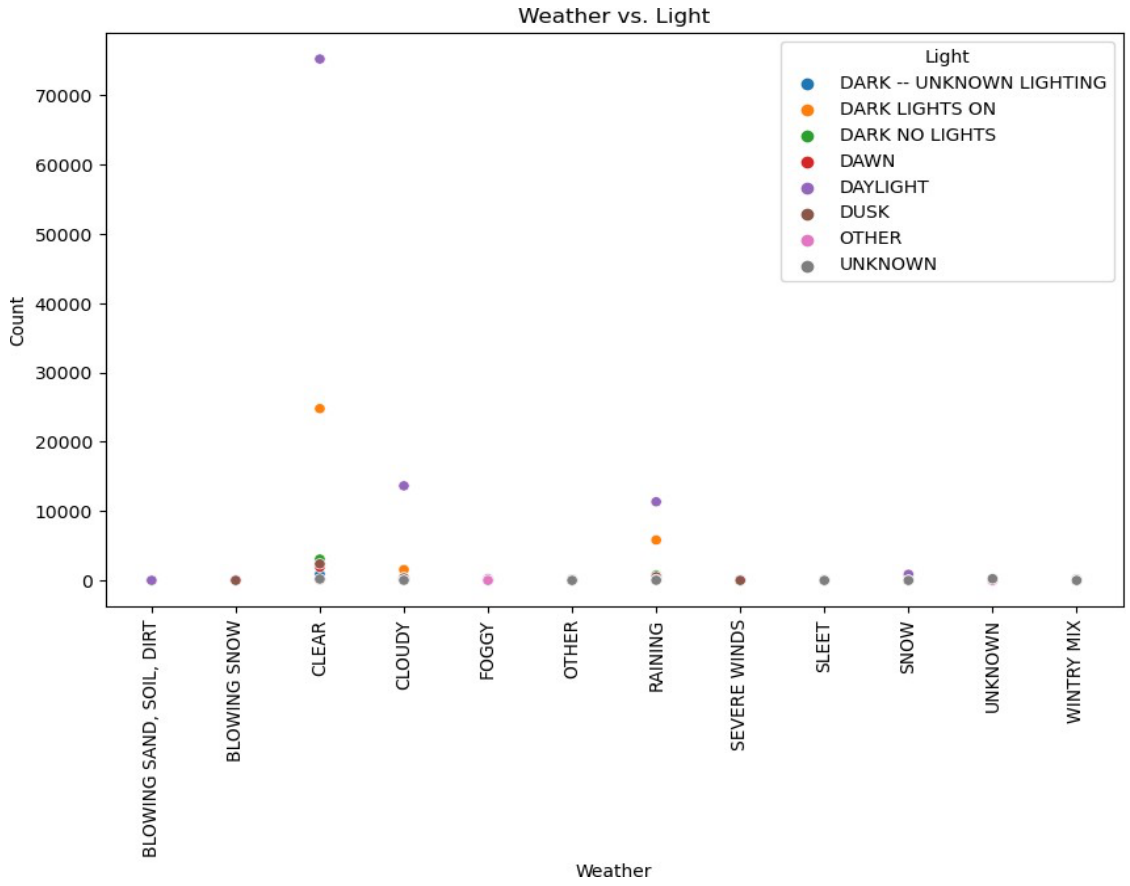

Figure 7

Figure 8 shows a line chart illustrating the relationship between vehicle speed and vehicle damage extent. The two vehicle damage extents are "disabling" and "destroyed". The chart indicates that as the speed increases, the level of destruction to the vehicle also increases. At lower speed limits (less than 15 miles per hour) and higher speed limits (around 65 miles per hour), the vehicle damage extent appears to be relatively low. This suggests that crashes occurring at these speed ranges result in less severe vehicle damage.

However, between speed limits of 20 to 50 miles per hour and speeds exceeding 70 miles per hour, the severity of vehicle damage is higher. This indicates that crashes at these higher speed ranges tend to result in more extensive vehicle damage.

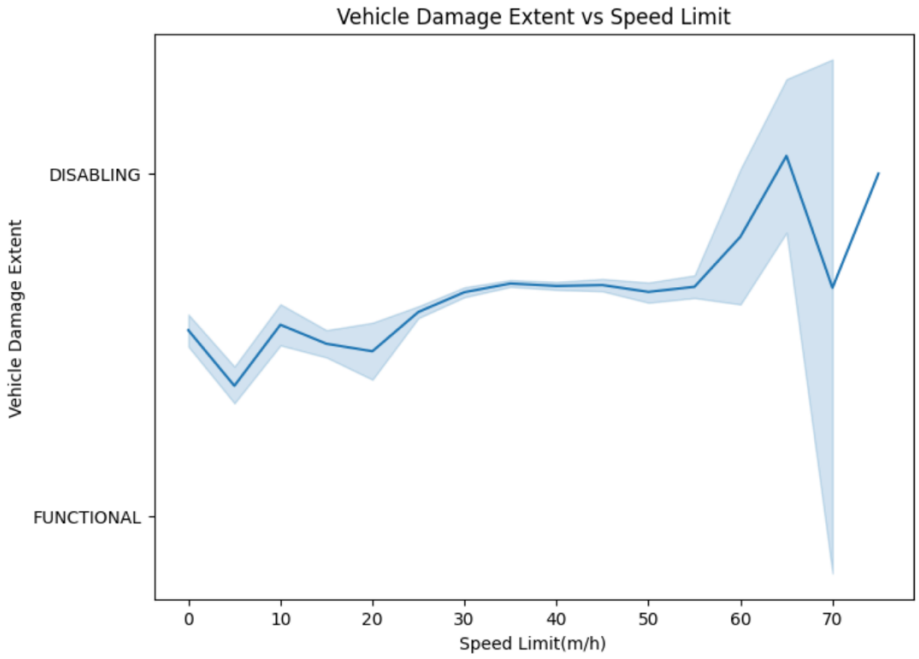

Figure 8

The Plotly library is used to create an interactive scatter mapbox visualization in the plot shown in Figures 9 and 10. By filtering the dataset, the plot focuses on the top five collision types. The scatter mapbox plot is created using Plotly Express, where latitude and longitude act as marker positions and collision type determines marker colors.

The map layout is personalized by selecting a style, zoom level, and center coordinates. Finally, this plot displays an interactive map of crash locations in Montgomery County, with different collision types represented by different colors.

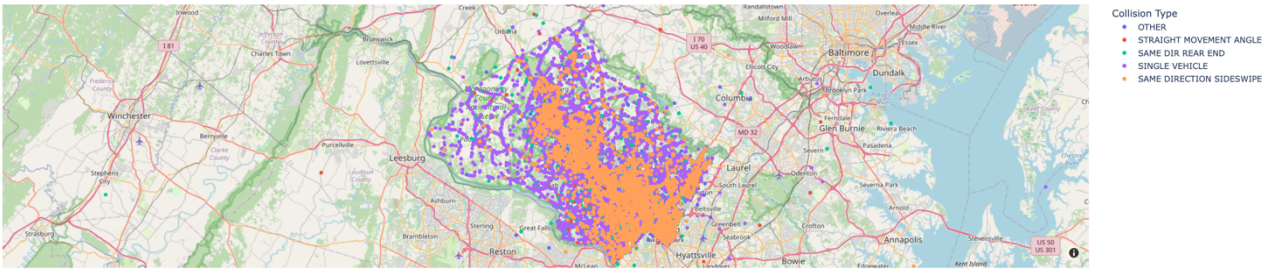

Figure 9

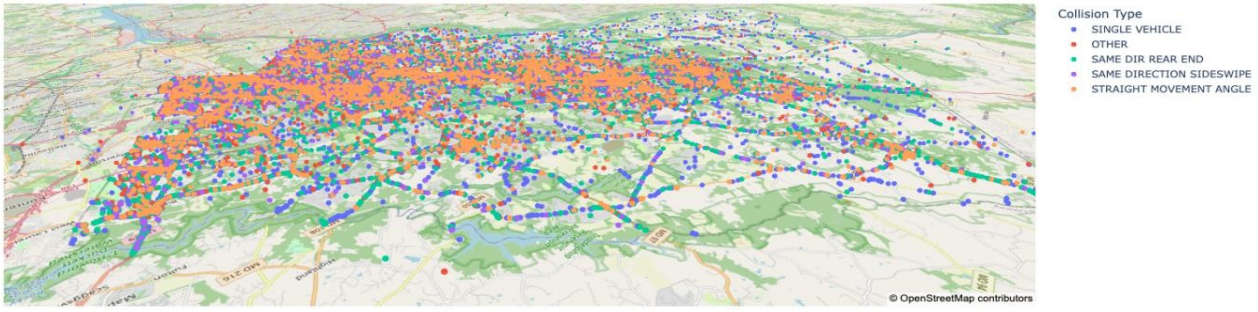

Figure 10

This chart, crafted using the Plotly library, visually represents the distribution of injury severities across different years. The initial step in the code involves converting the 'Crash Date/Time' field into datetime format and isolating the year. The time frame for the x-axis spans from the earliest to the latest year found in the data. A computation is performed to determine the frequency of each injury severity category on an annual basis through a combination of groupby and size functions. The graph is then configured to enable interactive analysis, with the inclusion of slider steps for annual data navigation.

For each subsequent year, a new segment of the bar graph is constructed to illustrate the frequency of injury severities, categorized on the x-axis and quantified on the y-axis, with data labels on the bars themselves. The graph's layout is meticulously defined, including titles for both axes and the chart itself, along with a slider in the update menu that permits users to sift through individual years or view the entire timeline simultaneously. The resulting visualization offers a user-friendly interface for examining and contrasting injury severity trends over time.

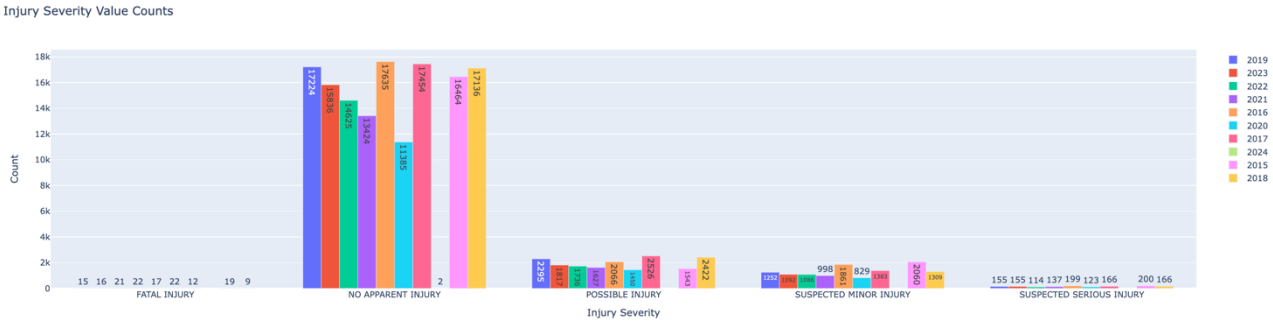

Figure 11

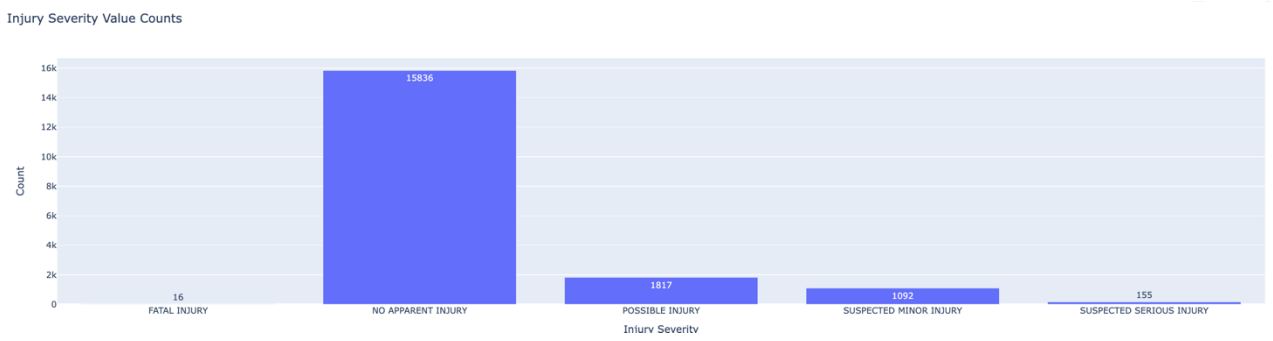

Figure 12

The table below shows the number of "Driver At Fault" incidents classified by "Vehicle Damage Extent." Analyzing the data, it is clear that the "Disabling" extent caused the greatest number of crashes, accounting for approximately 60,000 crashes. The "Functional" and "Superficial" extents are close behind, both at around 40,000 crashes.

In comparison, the extent of car destruction was responsible for approximately 7,000 crashes, while the "Unknown" damage extent was responsible for approximately 6,100 crashes. Finally, there were approximately 5,800 crashes in which no vehicle damage was reported.

Furthermore, when looking at the distribution of "Driver At Fault" incidents within each extent category, it is worth noting that incidents involving "Functional" and "Superficial" car damage have less than 50% driver fault, while the majority of other crashes have more than 60% driver fault. These numerical insights, when combined with the stacked bar chart, provide an in-depth understanding of the relationship between "Driver At Fault" incidents and various levels of "Vehicle Damage Extent."

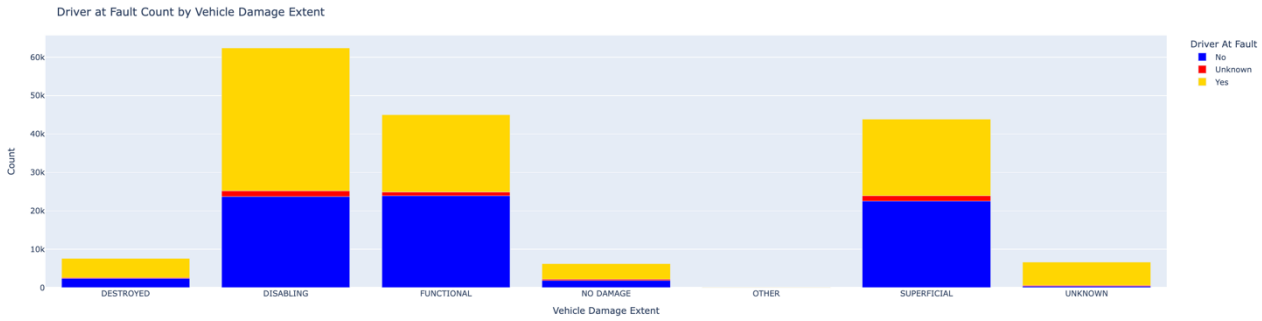

Figure 13

The graph depicts the number of "Driver At Fault" incidents classified by various types of "Light" conditions during crashes. Analyzing the data reveals that the majority of incidents occurred during "Daylight" hours, with a total count of 49,410. Following closely behind are incidents that occurred during "Dark Lights On" conditions, with 20,811 reported cases, and incidents that occurred during "Dusk" conditions, with 1,961 reported cases. The "Other" light condition, on the other hand, had the fewest incidents, with a total of 198.

When the distribution of "Driver At Fault" incidents within each light category is examined, it is clear that the majority of incidents across all light conditions (the "Yes" category) were attributed to drivers being at fault. Specifically, there were 57,154 incidents where the driver was at fault during "Daylight" conditions, compared to 4,941 incidents where the driver was not at fault. Similarly, there were 20,811 incidents with driver fault during "Dark Lights On" conditions, while 15,137 incidents were reported without driver fault.

Overall, the stacked line chart effectively visualizes the relationship between different light conditions and the distribution of "Driver At Fault" incidents, providing insights into how lighting affects driver behavior and fault attribution.

Figure 14

The graph displays information about various collision types, speed limits, and the corresponding counts. When the numbers are analyzed, we see that the collision type "ANGLE MEETS LEFT HEAD ON" had a count of 7 at a speed limit of 0, gradually increasing to 133 at a speed limit of 25, and then decreasing again.

Similarly, the collision type "ANGLE MEETS LEFT TURN" had a count of 12 when the speed limit was zero, peaking at 647 when the speed limit was 35, and then declining. The collision type "HEAD ON" had a count of 102 at a speed limit of 0, increasing to 827 at a speed limit of 35, and then decreasing. These trends hold true for other types of collisions, with varying counts at different speed limits. Overall, the graph sheds light on the relationship between collision types, speed limits, and occurrences.

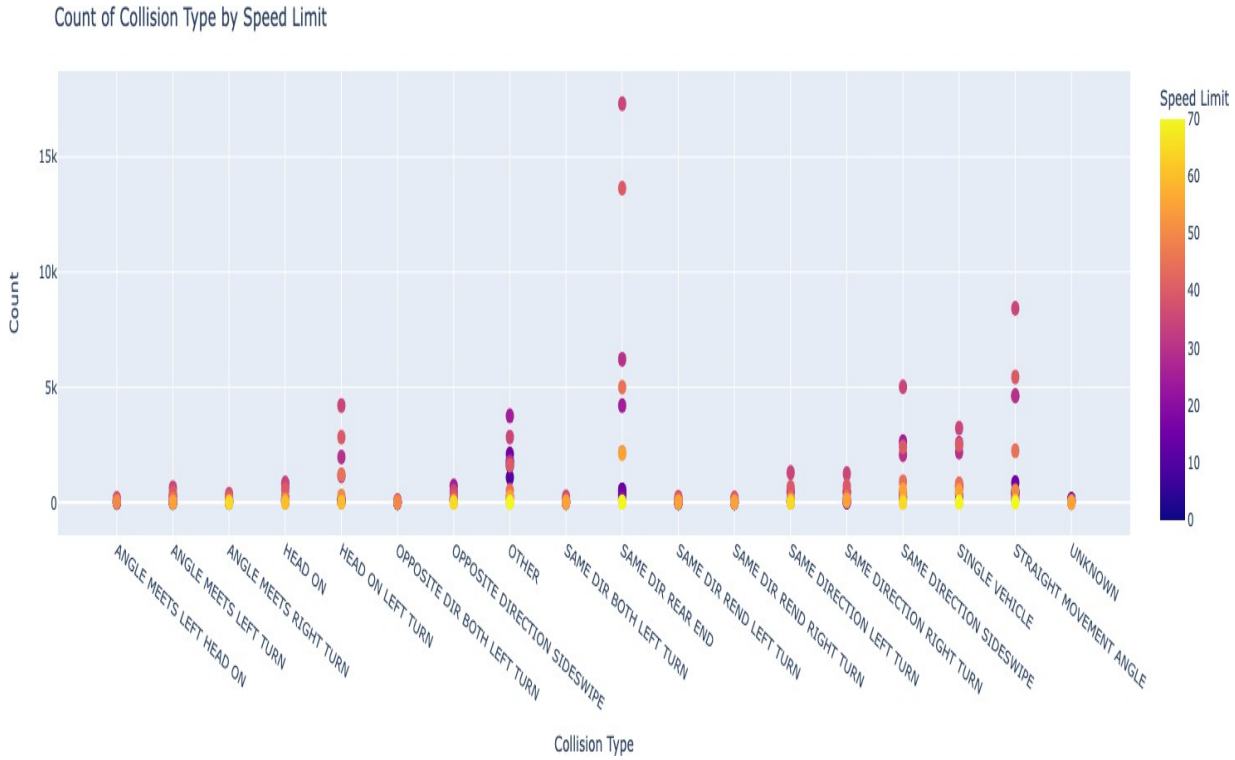

Figure 15

Data Cleaning:

To perform data cleaning, the null values present in the dataset are first analyzed. This is given in Figure 16. It can be seen that nearly 25 attributes contain null values.

```
Report Number                      0
Local Case Number                  0
Agency Name                        0
ACRS Report Type                   0
Crash Date/Time                    0
Route Type                     15928
Road Name                      15011
Cross-Street Type              15956
Cross-Street Name              15022
Off-Road Description          146695
Municipality                  143763
Related Non-Motorist          156594
Collision Type                   543
Weather                        12612
Surface Condition              18974
Light                           1343
Traffic Control                24198
Driver Substance Abuse         29439
Non-Motorist Substance Abuse  157660
Person ID                          0
Driver At Fault                    0
Injury Severity                    0
Circumstance                  132065
Driver Distracted By               0
Drivers License State           9098
Vehicle ID                         0
Vehicle Damage Extent            287
Vehicle First Impact Location    156
Vehicle Second Impact Location   256
Vehicle Body Type               2471
Vehicle Movement                 351
Vehicle Continuing Dir          2508
Vehicle Going Dir               2508
Speed Limit                        0
Driverless Vehicle                 0
Parked Vehicle                     0
Vehicle Year                       0
Vehicle Make                      22
Vehicle Model                     63
Equipment Problems             32588
Latitude                           0
Longitude                          0
Location                           0
Year                               0
dtype: int64
```

Figure 16

The 'Crash Date/Time' column contains both date and time of the car accidents together. Hence, they were split into two different columns namely ‘date’ and ‘time’. Later, the 'Crash Date/Time' column was dropped. The code that was used to perform this is shown in Figure 17.

```
# Converting the datetime column to datetime type
df['Crash Date/Time'] = pd.to_datetime(df['Crash Date/Time'])

# Splitting the datetime column into separate date and time columns
df['Date'] = df['Crash Date/Time'].dt.date
df['Time'] = df['Crash Date/Time'].dt.time
```

Figure 17

The attributes that are not required for the further analysis of the dataset such as 'Route Type', 'Crash Date/Time', 'Road Name', 'Cross-Street Type', 'Cross-Street Name', 'Off-Road Description', 'Municipality', 'Related Non-Motorist', 'Non-Motorist Substance Abuse', 'Circumstance','Drivers License State', 'Traffic Control','Vehicle First Impact Location', 'Vehicle Second Impact Location', 'Vehicle Continuing Dir', 'Vehicle Going Dir', 'Vehicle Make', 'Vehicle Model' were dropped using drop(). This is shown in Figure 18.

```
# Dropping all the unnecessary columns
df_1 = df.drop(['Route Type','Crash Date/Time','Road Name','Cross-Street Type','Cross-Street Name',
                'Off-Road Description','Municipality','Related Non-Motorist','Non-Motorist Substance Abuse',
                'Circumstance','Drivers License State', 'Traffic Control','Vehicle First Impact Location',
                'Vehicle Second Impact Location','Vehicle Continuing Dir',
                'Vehicle Going Dir','Vehicle Make','Vehicle Model'], axis=1)
df_1
```

Figure 18

Next, based on the predictor and target variables used for the models, the null values in the categorical columns are filled with either ‘UNKNOWN’ or ‘OTHER’ values. This can be seen in Figure 19.

```
# Selecting Predictor variable for filling null values
pred_list1 = [
    'Collision Type',
    'Light',
    'Vehicle Damage Extent',
    'Vehicle Movement',
    'Equipment Problems'
]
pred_list2 = ['Weather','Surface Condition','Driver Substance Abuse','Vehicle Body Type']
```

```
# Filling NaN values with specific values
df_1[pred_list1] = df_1[pred_list1].fillna('OTHER')
df_1[pred_list2] = df_1[pred_list2].fillna('UNKNOWN')
```

Figure 19

PREDICTIVE MODELS

For modelling, 'Injury Severity' is taken as the target variable as the business question deals with the injury levels. The attributes 'ACRS Report Type', 'Collision Type', 'Weather', 'Surface Condition', 'Light', 'Driver Substance Abuse', 'Driver At Fault', 'Driver Distracted By', 'Vehicle Damage Extent', 'Vehicle Body Type', 'Vehicle Movement', 'Speed Limit', 'Driverless Vehicle', 'Parked Vehicle', and 'Equipment Problems' are taken as the predictive variables.

Label encoding is performed for the predictors which was imported from the .py file as the function called apply_label_encoding. Another function named split_data was also included in the .py file to split the dataset into train and test. The code snippet from the .py file can be seen in Figure 20 and the functions called in the main script can be seen in Figure 21.

```
def apply_label_encoding(df, predictors):
    label_encoder = LabelEncoder()
    for feature in predictors:
        df[feature] = label_encoder.fit_transform(df[feature].astype(str))
    return df

def split_data(X, y, test_size=0.2, random_state=None):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)
    return X_train, X_test, y_train, y_test
```

Figure 20

The predictor variables (X) and the target variable (y), along with optional test size and random state parameters, are inputs to the train_test_split function.
X stands for the characteristics or predictor variables that will be employed in the prediction process.
The variable we want to predict or categorize is the target variable, represented by y.
The percentage of the dataset that should be allotted to the testing set is specified by the test_size parameter. 0.2 in this instance means that 20% of the data will be used for testing.

The split is made reproducible by using the random_state parameter. It can be set to a specific number, like 95, to ensure that the same split is produced each time the code is executed.

The data is then divided into the four subsets X_train, X_test, y_train, and y_test by the train_test_split function. These subsets, for the predictor variables and target variable, respectively, represent the training and testing data.

The predictor variables used in model training are contained in X_train.
The predictor variables for assessing the model's performance on unobserved data are contained in X_test.
The target variable values for the training set are contained in y_train.
The values of the target variables for the testing set are contained in y_test.
By dividing the data into training and testing sets, it is possible to train models on the training set and evaluate their performance and generalizability on the testing set.

```
# Calling label Encoding function
from Function_FinalP import apply_label_encoding
df_1 = apply_label_encoding(df_1, predictors)
```

```
# Calling function for Splitting the data into Train and Test
from Function_FinalP import split_data
X = df_1[predictors]
y = df_1[target]
X_train, X_test, y_train, y_test = split_data(X, y, test_size=0.2, random_state=95)
```

Figure 21

Gradient Boosting Classifier:

The first model used for prediction is the Gradient Boosting Classifier. It is an ensemble learning technique that combines various weak learners (decision trees) to produce a powerful predictive model. It operates by fitting new models repeatedly to the residuals of the prior models, concentrating on the occurrences that were predicted erroneously. This cycle repeats until a predetermined model count is reached or until a predetermined level of performance is attained.

Figure 22 represents the results of the gradient boosting classifier model. It states that the accuracy of the model is 84%. Interpreting the accuracy of 84% means that the model correctly predicted the injury severity in approximately 84% of the cases. It indicates that the model is performing reasonably well in classifying the instances into their respective injury severity categories.

Gradient Boosting Classifier Results:

	precision	recall	f1-score	support
FATAL INJURY	0.00	0.00	0.00	36
NO APPARENT INJURY	0.89	0.97	0.93	28178
POSSIBLE INJURY	0.41	0.34	0.37	3567
SUSPECTED MINOR INJURY	0.39	0.09	0.15	2369
SUSPECTED SERIOUS INJURY	0.23	0.04	0.07	271
accuracy			0.84	34421
macro avg	0.38	0.29	0.30	34421
weighted avg	0.80	0.84	0.81	34421

Figure 22

FATAL INJURY: The model's precision was 0, meaning that no events that were predicted to result in "FATAL INJURY" really did so. The harmonic mean of precision and recall yields an accurate assessment of the model's performance for this class with a f1-score of 0.

NO APPARENT INJURY: Based on the precision of 0.89, 89% of the predictions for "NO APPARENT INJURY" was accurate. Recall of 0.97 indicates that 97% of the actual instances of

"NO APPARENT INJURY" was correctly detected by the model. The high f1-score of 0.93 indicates an excellent performance in class prediction.

POSSIBLE INJURY: This class's precision is 0.41, meaning that 41% of predictions for "POSSIBLE INJURY" were accurate. The model correctly detected 34% of the actual incidents of "POSSIBLE INJURY" with a recall of 0.34. The f1-score of 0.37 indicates a fair level of performance in class prediction.

SUSPECTED MINOR INJURY: The accuracy for this class is 0.39, which means that 39% of the predictions for "SUSPECTED MINOR INJURY" were accurate. Recall of 0.09 indicates that only 9% of the real "SUSPECTED MINOR INJURY" events were accurately detected by the model. The low f1-score of 0.15 indicates a weak performance in class prediction.

SUSPECTED SERIOUS INJURY: This class's accuracy is 0.23, meaning that 23% of predictions for "SUSPECTED SERIOUS INJURY" were accurate. The recall of 0.04 indicates that only 4% of the actual incidents of "SUSPECTED SERIOUS INJURY" were properly detected by the model. The extremely low f1-score of 0.07 indicates poor performance in class prediction.

With excellent precision, recall, and f1-score, the model performs well overall in predicting "NO APPARENT INJURY" cases. The prediction of additional injury severity classes, notably "SUSPECTED MINOR INJURY" and "SUSPECTED SERIOUS INJURY," where the recall is especially low, is a challenge. This suggests that the model might have trouble accurately differentiating these classes. The model's performance may still need to be improved for these particular classes.

Random Forest Classifier:

The next model used is a random forest classifier. It is a machine learning technique that is frequently used for classification applications. It is a component of ensemble learning techniques, which combine various independent models to produce predictions.
During the training phase, the random forest classifier builds a large number of decision trees. A random subset of the training data and a random subset of the characteristics are used to construct each decision tree. The random forest becomes more robust and less prone to bias as a result of the randomness's role in introducing variety among the trees and lowering overfitting.

The results of the random forest classifier model are shown in Figure 23. It reports that the model's accuracy is 83%. The accuracy of 83% can be understood to suggest that in about 83% of the cases, the model accurately anticipated the severity of the injury. It shows that the model does a fair job of classifying the cases into the appropriate injury severity groups.

Random Forest Classifier Results:

	precision	recall	f1-score	support
FATAL INJURY	0.75	0.67	0.71	36
NO APPARENT INJURY	0.91	0.94	0.92	28178
POSSIBLE INJURY	0.38	0.35	0.36	3567
SUSPECTED MINOR INJURY	0.29	0.20	0.24	2369
SUSPECTED SERIOUS INJURY	0.18	0.08	0.11	271
accuracy			0.82	34421
macro avg	0.50	0.45	0.47	34421
weighted avg	0.80	0.82	0.81	34421

Figure 23

FATAL INJURY: With a precision of 0.75, the model correctly predicted "FATAL INJURY" in 71% of the situations. The recall of 0.67 indicates that 67% of the actual incidents of "FATAL INJURY" were correctly detected by the model. The f1-score of 0.71 indicates that the prediction of this class was performed rather well.

NO APPARENT INJURY: According to the precision of 0.91, 91% of the predictions for "NO APPARENT INJURY" was accurate. Recall of 0.94 indicates that 94% of the real "NO APPARENT INJURY" cases were accurately detected by the model. The high f1-score of 0.92 denotes an excellent performance in class prediction.

POSSIBLE INJURY: This class's precision is 0.38, meaning that 38% of predictions for "POSSIBLE INJURY" were accurate. A recall of 0.35 indicates that 35% of the actual occurrences of "POSSIBLE INJURY" were correctly detected by the model. The f1-score of 0.36 indicates a fair level of performance in class prediction.

SUSPECTED MINOR INJURY: This class's precision is 0.29, meaning that 29% of predictions for "SUSPECTED MINOR INJURY" were accurate. Only 20% of the actual incidents of "SUSPECTED MINOR INJURY" were accurately detected by the model, according to the recall of 0.20. The 0.24 f1-score indicates a pretty poor performance in class prediction.

SUSPECTED SERIOUS INJURY: This class's precision is 0.18, meaning that 18% of predictions for "SUSPECTED SERIOUS INJURY" were accurate. The recall of 0.08 indicates that only 8% of the actual incidents of "SUSPECTED SERIOUS INJURY" were properly identified by the model. The f1-score of 0.11 indicates a subpar performance in class prediction.

With high precision, recall, and f1-score, the model does a good job at predicting "NO APPARENT INJURY" cases altogether. The prediction of additional injury severity classes, notably "SUSPECTED MINOR INJURY" and "SUSPECTED SERIOUS INJURY," where the recall is especially low, is a challenge. This suggests that the model might have trouble accurately differentiating between these classes. The model's performance may still need to be improved for these particular classes.

INTERPRETATION OF THE MODELS

The comparison between the Gradient Boosting Classifier and the Random Forest Classifier reveals differences in their predictive success for injury severity in car accidents. The Gradient Boosting Classifier slightly edges out with an accuracy of 84%, compared to the Random Forest Classifier's 83%.

When focusing on precision, which assesses the correct positive predictions, the Gradient Boosting Classifier generally shows superior results across most categories of injury severity. However, it's important to note that both models exhibit considerable challenges in accurately predicting the categories "SUSPECTED MINOR INJURY" and "SUSPECTED SERIOUS INJURY," resulting in lower precision scores for these injuries.

Looking at the recall metric, which evaluates the detection of true positive instances, the Gradient Boosting Classifier consistently demonstrates higher recall values across the majority of injury severity classes compared to the Random Forest Classifier. This suggests that the former is more adept at identifying actual cases in each category. Nonetheless, both models struggle to correctly detect "SUSPECTED MINOR INJURY" and "SUSPECTED SERIOUS INJURY," which is reflected in their lower recall scores for these categories.

Considering the F1-score, which is a harmonic mean of precision and recall, the Gradient Boosting Classifier again shows a better performance for most categories. However, the models perform similarly on the more challenging "SUSPECTED MINOR INJURY" and "SUSPECTED SERIOUS INJURY" classes, with both exhibiting lower F1-scores, indicating difficulties in accurate prediction.

Overall, while both algorithms have their limitations, particularly with the challenging task of forecasting “SUSPECTED MINOR INJURY” and “SUSPECTED SERIOUS INJURY,” the Gradient Boosting Classifier tends to have a slight advantage in overall performance across the metrics of precision, recall, and F1-score for this specific dataset.

REFERENCES

Kazarinoff, P. D. (n.d.). *Calling Functions from Other Files - Problem Solving with Python*. Calling Functions From Other Files - Problem Solving With Python. https://problemsolvingwithpython.com/07-Functions-and-Modules/07.05-Calling-Functions-from-Other-Files/

Crash Reporting - Drivers Data - Catalog. (n.d.). Crash Reporting - Drivers Data - Catalog. https://catalog.data.gov/dataset/crash-reporting-drivers-data

sklearn.ensemble.GradientBoostingClassifier. (n.d.). Scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Plotly. (n.d.). Plotly Python Graphing Library. https://plotly.com/python/

sklearn.ensemble.RandomForestClassifier. (n.d.). Scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html