Predictive Analysis of Traffic Crashes in Chicago: Crash Types and Damage Estimation

1st Prathibha Vuyyala
Engineering Science - Data Science
University at Buffalo
Buffalo, US
pvuyyala@buffalo.edu

2nd Tharun Teja Mogili
Engineering Science - Data Science
University at Buffalo
Buffalo, US
tharunte@buffalo.edu

3rd Pavan Pajjuri
Engineering Science - Data Science
University at Buffalo
Buffalo, US
spajjuri@buffalo.edu

***Abstract*—This analysis of Chicago traffic accident data from June to December 2023 focuses on data cleaning, validation, and standardization. Missing values were imputed, duplicates and inconsistencies addressed, and categorical features encoded for machine learning. The cleaned dataset is prepared for further statistical analysis of traffic patterns.**

I. Introduction

An average span of four days, Chicago can record up to over a thousand car accidents. When you include drivers, passengers, pedestrians and cyclists, up to two thousand people can be effected. Forty-five percent of the people will experience a minor to fatal injury.

Traffic accidents represent a critical challenge for urban safety management, resulting in injuries, fatalities, and economic loss. This project focuses on leveraging the Chicago crashes dataset to address specific problems related to traffic crashes

II. Problem Statement

A. Problem Statement Questions

- **Predicting Crash Type:** Utilizing available data features, we aim to develop a predictive model that can accurately classify the crash type (Injury or No Injury) based on factors such as Road conditions, weather conditions, and traffic control devices. Understanding if the crash could injurious can inform targeted prevention strategies and resource allocation.
- **Identifying Risk Factors for Severe Accidents:** We will investigate how various elements (e.g., environmental conditions, traffic volume) correlate with severe accidents. This analysis aims to uncover actionable insights that can guide interventions aimed at reducing the occurrence and impact of high-severity crashes.

B. Background of the Problem

Traffic accidents in urban environments, such as Chicago, are a persistent concern, contributing to injuries, fatalities, and substantial economic losses. With the increasing complexity of urban traffic systems and the multitude of factors influencing crash events—ranging from road conditions, weather, and traffic control devices to driver behavior—the need for predictive tools has grown. This project aims to explore these complex dynamics by utilizing the Chicago crashes dataset, which provides a wealth of structured data that can be used to uncover patterns and factors contributing to traffic accidents. By focusing on predicting crash types, estimating damage severity, and identifying risk factors for severe accidents, this project addresses key challenges in urban traffic safety management. Furthermore, the project seeks to better understand the conditions under which severe injuries and hit-and-run incidents occur, and how poor roadway surface conditions impact accident rates.

C. Significance of the Problem and Project Contribution

This project is significant because traffic accidents have far-reaching impacts on public safety, economic costs, and emergency services. The ability to predict crash types and estimate damage severity could lead to more efficient allocation of resources, such as emergency services and law enforcement. Moreover, identifying the conditions that lead to severe accidents enables targeted interventions, such as improving road infrastructure or adjusting traffic control measures in high-risk areas. Understanding trends in hit-and-run incidents and the effects of road conditions can guide policy decisions aimed at enhancing traffic safety and accountability. By providing actionable insights through data-driven models, this project has the potential to improve urban traffic management strategies, reduce accident rates, and mitigate the impacts of crashes on public health and safety.

D. Potential Contribution of the Project

This project aims to contribute significantly to the domain of urban traffic safety by developing data-driven models that can predict crash types and estimate damage severity. By leveraging predictive analytics, this project will help inform targeted strategies for accident prevention, particularly in identifying conditions under which crashes are more likely to result in injuries. This predictive capability can enable city authorities, transportation agencies, and policymakers to allocate resources where they are most needed, reducing response times for emergency services and guiding future infrastructure investments.

Moreover, by estimating damage severity, the project will offer valuable insights for planning and improving road safety.

Understanding which factors—whether environmental, vehicular, or road-related—contribute to more severe crashes can help refine traffic regulations, implement better warning systems, and design safer intersections. The analysis of hit-and-run incidents can also reveal important patterns that inform policy decisions to enhance public safety, hold offenders accountable, and provide better post-crash support for victims. Additionally, investigating how roadway conditions influence crash rates and severity offers actionable information to guide road maintenance and policy reforms, which are crucial for reducing accident rates in cities with challenging environmental conditions like Chicago. Ultimately, this project will empower stakeholders to proactively manage traffic safety, contributing to a safer, more sustainable urban transport system.

III. Data Sources

The primary dataset for this project comes from the Chicago Traffic Crashes Dataset, which is publicly available through the City of Chicago's data portal. This dataset contains detailed information about traffic accidents that have occurred in the city of Chicago, with over 870,000 records spanning multiple years, and includes variables such as crash type, weather conditions, road surface conditions, and traffic control device statuses.

The data from the Chicago Traffic Crashes dataset is comprehensive and includes numerous variables that allow for an in-depth analysis of crash incidents. With over 870,000 rows and 48 columns, this dataset is well-suited for addressing the project's core objectives of predicting crash types, estimating damage severity, and identifying risk factors for severe accidents. This data source ensures that the dataset is both large enough to yield significant insights and contains the necessary variety of features to facilitate predictive modeling and exploratory data analysis. The dataset from the Chicago site spans from 2013 to the present. For the purposes of this analysis, we will focus exclusively on a six-month period, specifically from June 2023 to December 2023

You can access the data source at Data Source.

IV. Data Cleaning/ Processing

Effective data cleaning and processing is essential to ensure the accuracy and reliability of the analysis. The Chicago Traffic Crashes dataset, like most real-world datasets, contains inconsistencies, missing values, and redundant information that must be addressed. Below is an outline of the 10 distinct processing and cleaning steps applied to the dataset:

A. Dropping Null Rows and Columns

The dataset contains records with missing values, prompting the need for a structured approach to manage these gaps. A column threshold of 30% was set, leading to the removal of columns with excessive missing data. Consequently, the following columns were dropped: *WORK_ZONE_TYPE, NOT_RIGHT_OF_WAY_I, STATEMENTS_TAKEN_I, PHOTOS_TAKEN_I, WORK_ZONE_I, CRASH_DATE_EST_I, WORKERS_PRESENT_I, LANE_CNT, DOORING_I, INTERSECTION_RELATED_I*. This process reduced the dataset from 48 columns to 38 columns. Additionally, a row threshold of 20% was implemented to eliminate rows with excessive missing values, ensuring the quality of the remaining data and minimizing potential bias in subsequent analyses. Additionally, a row threshold of 20% was implemented to eliminate rows with excessive missing values, ensuring the quality of the remaining data and minimizing potential bias in subsequent analyses.

B. Handling Missing Data

The dataset was analyzed to identify missing values in both categorical and numerical columns. For categorical variables such as `REPORT_TYPE`, `HIT_AND_RUN_I`, and `MOST_SEVERE_INJURY`, missing values were imputed with the mode of each column to maintain consistency. The mode values used were:

- `REPORT_TYPE`: "NOT ON SCENE (DESK REPORT)"
- `HIT_AND_RUN_I`: "Y"
- `MOST_SEVERE_INJURY`: "NO INDICATION OF INJURY"

For numerical variables, missing values were also filled using the mode. Key columns with missing values included:

- `INJURIES_TOTAL`
- `INJURIES_FATAL`
- `INJURIES_INCAPACITATING`
- `INJURIES_NON_INCAPACITATING`
- `INJURIES_REPORTED_NOT_EVIDENT`
- `INJURIES_NO_INDICATION`
- `INJURIES_UNKNOWN`
- `LATITUDE`
- `LONGITUDE`

C. Capping Outliers

To address the issue of potential outliers in the `INJURIES_NO_INDICATION` column, a threshold was established to cap values exceeding 10. Any record with `INJURIES_NO_INDICATION` greater than 10 was adjusted to a maximum value of 10. This capping method helps mitigate the influence of extreme values on the analysis, ensuring that the dataset remains robust for further statistical evaluations.

D. Standardization of Numerical Features

The numerical features in the dataset underwent standardization to ensure that they have a mean of 0 and a standard deviation of 1. This process was applied to all numeric columns identified in the dataset. The `StandardScaler` from scikit-learn was utilized for this purpose, transforming the data accordingly. Standardization is crucial for many machine learning algorithms as it enhances the model's performance by ensuring that each feature contributes equally to the distance calculations. A descriptive summary of the standardized features numerical features was printed for further analysis in the code.

E. Dealing with Categorical Values and Inconsistency - Mapping

The dataset contains several categorical variables that may have inconsistent entries, which could affect the analysis. A systematic mapping approach was employed to standardize the values in key categorical columns, ensuring uniformity. The following steps were taken:

- Traffic Control Device: Multiple entries were consolidated into broader categories (e.g., “TRAFFIC SIGNAL”, “FLASHING CONTROL SIGNAL”, and “RAILROAD CROSSING GATE” were all mapped to “SIGNAL”). This mapping reduced variability and enhanced the clarity of data, resulting in a more manageable distribution of categories.
- Weather Conditions: Various weather descriptions were similarly standardized to ensure consistency (e.g., “RAIN” includes “FREEZING RAIN/DRIZZLE”). This allows for better comparison and analysis of accident occurrences under different weather conditions.
- Primary Contributory Cause: The mapping transformed numerous specific causes of crashes into broader categories (e.g., “FAILING TO YIELD RIGHT-OF-WAY” became “YIELDING ISSUES”). This aids in summarizing the data while still retaining essential information for analysis.

The resulting distributions of the standardized categories were examined, revealing clearer patterns and relationships among the variables, which are crucial for subsequent analysis.

F. Data Type Change

The `DATE_POLICE_NOTIFIED` column was converted to a datetime format using `pd.to_datetime()`, allowing for efficient date manipulation and analysis. The format specified was `%m/%d/%Y %I:%M:%S %p`, and any errors during conversion were handled by coercing them to `NaT`.

Several columns related to the timing of crashes were transformed to the category data type to optimize memory usage and improve performance during analysis. These columns included:

- `CRASH_HOUR:` Representing the hour of the crash.
- `CRASH_DAY_OF_WEEK:` Indicating the day of the week the crash occurred.
- `CRASH_MONTH:` Denoting the month in which the crash happened.

By converting these columns to categorical types, the dataset is better structured for analysis and visualization tasks that may benefit from treating these values as distinct categories.

G. Redundant Columns Removal

Certain columns identified as redundant were removed from the dataset to streamline the data and enhance its utility for analysis. These columns were deemed of little use either due to their limited analytical value or because their information was already captured in other columns. The columns removed include:

- `LOCATION`: Contains information that is duplicative or less relevant.
- `SEC_CONTRIBUTORY_CAUSE`: Provides minimal additional insights beyond the primary cause.
- `STREET_DIRECTION`: Does not significantly contribute to the analysis.
- `STREET_NAME`: Offers little additional context for the analysis being conducted.
- `STREET_NO`: Similar to `STREET_NAME`, it does not add meaningful information.

This cleaning step helps focus on the most relevant data for further analysis and model building.

H. Temporal Consistency Checks

- The check for logical consistency revealed that all accident records are correctly timestamped, with no instances where the `CRASH_DATE` is later than the `DATE_POLICE_NOTIFIED`.
- This indicates that all reported accidents were notified appropriately without any discrepancies in the temporal order of events.
- The analysis found a significant number of duplicate crash records, totaling 34,419 occurrences across various `CRASH_DATE` entries.
- Specifically, there are 11,996 unique dates with multiple records, indicating that numerous accidents were reported on the same date and time.
- This could be due to multiple accidents occurring simultaneously or errors in data entry.

I. Cross-Validation of Related Columns

- The validation check for the relationship between `INJURIES_TOTAL` and `MOST_SEVERE_INJURY` found 0 invalid records. This means that all records align logically; when there are injuries reported (`INJURIES_TOTAL` ¿ 0), the severity classification does not mistakenly indicate ”NO INDICATION OF INJURY.”
- The cross-validation of the relationship between `WEATHER_CONDITION` and `LIGHTING_CONDITION` identified 230 records where the weather was reported as ”SNOW” while the lighting condition was classified as ”DAYLIGHT.” This inconsistency suggests a potential data quality issue, as snowy conditions typically occur when there is limited daylight, indicating possible errors in reporting or categorization.

J. Categorical Encoding

- Ordinal Encoding: Specific columns with a natural order were transformed using ordinal encoding to convert categorical values into numerical representations. The mappings for each column were defined as follows:
  - `Lighting Condition:` Values were assigned from 1 (*DAYLIGHT*) to 6 (*UNKNOWN*).

- Most Severe Injury: Injury severity was mapped from 1 (*NO INDICATION OF INJURY*) to 5 (*FATAL*).
- Report Type: Categorical values were encoded to 1 (*NOT ON SCENE*) and 2 (*ON SCENE*).
- Crash Type: Encoded as 1 (*NO INJURY*) and 2 (*INJURY/TOW*).
- Hit and Run Indicator: Mapped to binary values, where ’Y’ is 1 and ’N’ is 0.
- Damage: Categorized into three levels, from 1 (*OVER $1,500*) to 3 (*$500 OR LESS*).

- Nominal Encoding: For columns that do not have a natural order, one-hot encoding was applied. This method created binary columns for each category within the specified nominal columns:
  - Traffic Control Device
  - Device Condition
  - Weather Condition
  - First Crash Type
  - Trafficway Type
  - Alignment
  - Roadway Surface Condition
  - Road Defect
  - Primary Contributory Cause

V. Exploratory Data Analysis

1) Summary Statistics: The histograms and boxplots provide valuable insights into the distribution of various numerical features in the dataset, as well as potential outliers.

From the histograms, it’s clear that variables such as posted speed limit, lane count, and number of units involved in crashes have skewed distributions, with most crashes occurring at lower speed limits (under 40 mph), involving fewer lanes and fewer vehicles. The histogram for crash hour shows that crashes are more frequent during the middle of the day, with a peak between 12 PM and 6 PM. Other variables, like injury-related counts, display a concentration of data around lower values, indicating that the majority of crashes result in minimal or no injuries.

The boxplots highlight outliers in some features, particularly in variables such as street number and number of units, where extreme values extend beyond the whiskers. These outliers could represent crashes in unique locations or incidents involving multiple vehicles. For instance, the beat of occurrence shows variability with a noticeable range of values, but extreme cases still occur beyond the normal distribution. The presence of outliers in injury-related variables is less prominent, but outliers in date differences suggest that some incidents take significantly longer to be reported than others. These insights suggest that while the majority of crashes follow expected patterns, there are exceptions that may warrant further investigation, particularly those involving delayed reporting or unusual locations.

2) Binning Hours to Analyze Distribution of Accidents: The distribution of crashes by hour bins reveals several important patterns. The data shows that the highest number of crashes occurs between 12 PM and 6 PM, likely corresponding to peak traffic periods such as lunch breaks and afternoon commutes. This suggests that increased traffic volume, coupled with potentially higher stress levels during these hours, plays a role in the elevated crash rates. The 6 AM to 12 PM time bin also shows a considerable number of crashes, which could be attributed to morning rush hours as people head to work or school, indicating another peak period for traffic accidents.

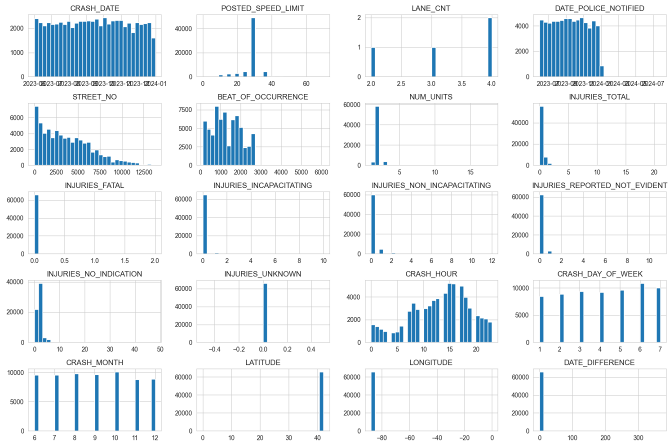

Fig. 1. Data Distribution of Features

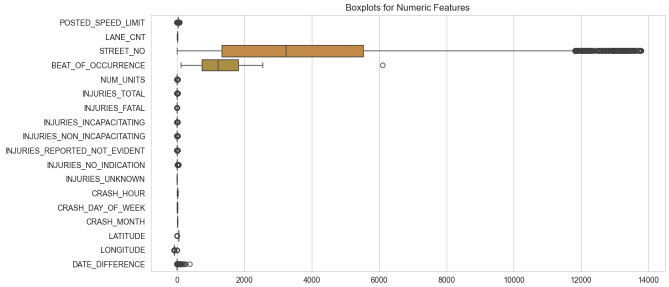

Fig. 2. Box plot for numeric features

In contrast, significantly fewer crashes are observed between midnight and 6 AM, which is expected due to reduced traffic volume during late-night hours. The crash rate between 6 PM and midnight is lower than in the afternoon but remains substantial, likely due to evening activities and post-work commutes. These findings suggest that targeted interventions, such as increased traffic monitoring or public safety campaigns, may be most effective when implemented during afternoon and morning rush hours, where the risk of crashes appears to be the highest.

3) Geographical Distribution of Accidents: The scatter plot visualizing the geographical distribution of accidents provides significant insights into the spatial concentration of crashes. The dense clustering of points indicates that accidents are highly concentrated in specific regions, which may correspond to urban areas with high traffic volume. These clusters can help identify potential accident-prone zones, possibly due to a combination of factors such as road design, traffic congestion,

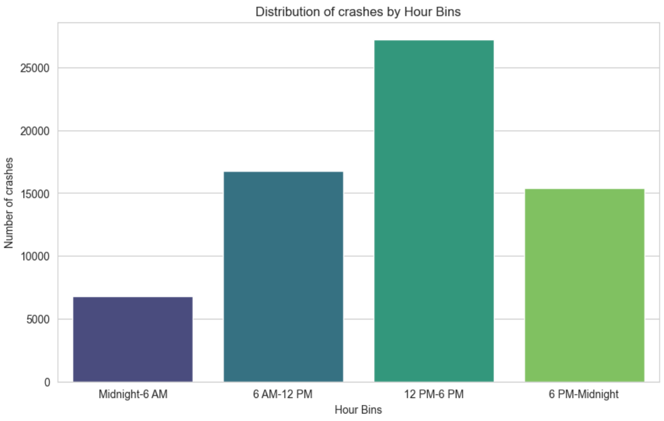

Fig. 3. Distribution of crashes by Hour Bins

and the presence of key intersections or landmarks. Urban planners and local authorities can use this data to focus on improving road safety in these regions, perhaps by installing better traffic control devices or redesigning intersections. Furthermore, the spread of points outside the dense areas suggests that accidents also occur in less populated or suburban areas, though at a much lower frequency. By analyzing these patterns further, transportation officials can determine whether these accidents are due to factors like road conditions, inadequate lighting, or speeding. This geographical analysis is a crucial component in identifying high-risk areas and implementing targeted interventions to reduce the occurrence of accidents in both densely populated urban areas and less traveled suburban regions.

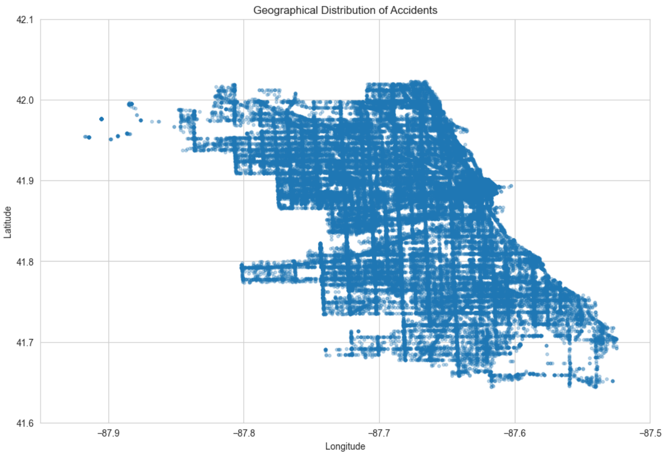

Fig. 4. Geographical Distribution of Accidents

4) Traffic Accidents by Months: The line plot showing traffic accidents by month for the year 2023 highlights noticeable fluctuations in the number of accidents throughout the year. A peak in accidents can be observed in October, with a slight decline following it. This suggests that there could be seasonal factors, such as changes in weather or traffic volume, contributing to a higher accident rate during this period. In contrast, there is a sharp drop in accidents in December, which may be due to holidays or different traffic patterns during the winter months, resulting in reduced road activity or heightened caution. Interestingly, August and November also demonstrate higher accident rates, possibly due to transitional periods between summer and fall, where weather conditions might vary, leading to unpredictable road safety. This analysis can help city planners and authorities identify specific months when increased road safety measures should be implemented to mitigate the risk of traffic accidents, particularly during peak months like October and transitional months like August and November.

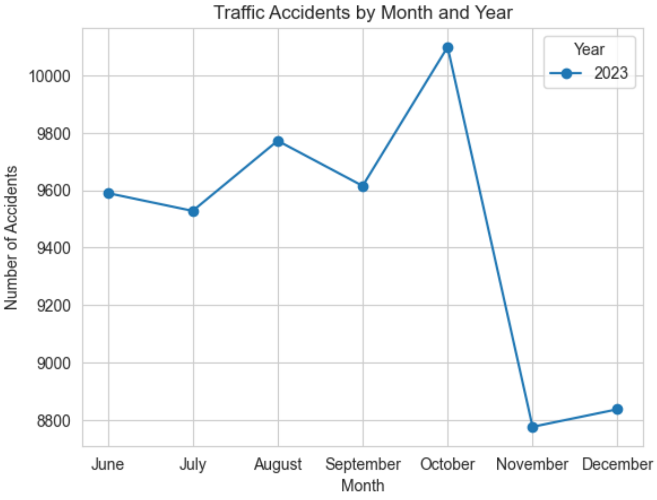

Fig. 5. Traffic Accidents by Months

5) Total Accidents by Primary Cause of Traffic Accidents: The bar plot visualizing the primary causes of traffic accidents reveals some significant insights. The leading cause of accidents is failing to yield the right-of-way, accounting for 7,578 incidents. This suggests that right-of-way violations remain a persistent issue on the roads, possibly due to lack of awareness or disregard for traffic rules. Another major contributor is following too closely, with 5,469 incidents, which may indicate a prevalent issue with tailgating behavior, especially in high-traffic areas. These two causes highlight the need for stricter enforcement and education around defensive driving techniques. An interesting observation from the chart is that a substantial number of accidents (27,734 cases) are marked as unable to determine the cause. This ambiguity suggests either inadequate data collection or challenges in pinpointing the specific cause of accidents, signaling an opportunity for better reporting mechanisms. Additionally, other causes, such as improper lane usage and disregarding traffic signals, also contribute significantly, underscoring the complexity of traffic safety issues that need to be addressed through both infrastructure improvements and driver education programs

6) Weather and Lighting Conditions: The heatmap displaying accidents based on weather and lighting conditions reveals some clear trends. Clear weather and daylight conditions

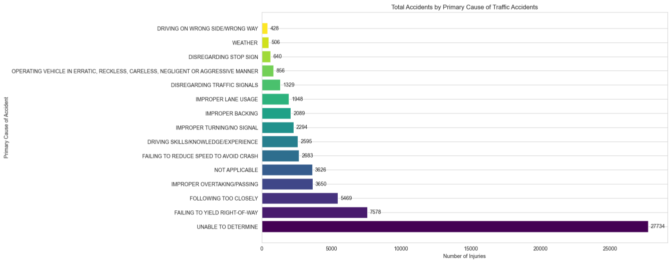

Fig. 6. Total Accidents by Primary Cause of Crash

show the highest number of accidents, with 35,206 accidents recorded under these conditions. This is likely due to the fact that most road activity happens during clear and daylight conditions, resulting in more opportunities for crashes despite the favorable weather and visibility. Similarly, clear weather combined with lighted roads also has a significant number of accidents, suggesting that despite better road conditions, other factors such as traffic density might play a role in accident occurrences. On the other hand, severe weather conditions like fog, snow, and rain under low visibility conditions (such as darkness or dawn) show relatively fewer accidents. This could imply that drivers are more cautious during these conditions or that there are fewer vehicles on the road. However, the risks posed by such weather should not be underestimated, as the combination of rain and darkness still results in a notable number of accidents (2,434 cases). This suggests that while extreme weather reduces the number of vehicles on the road, it increases the danger for those who are driving, necessitating targeted interventions like weather advisories or road safety campaigns.

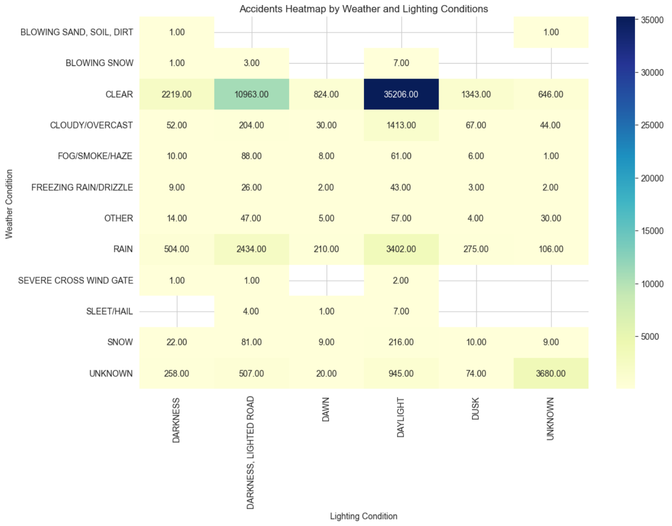

Fig. 7. Crashes Heatmap by Weather and Lighting Conditions

7) Distribution of Crash Types Across Different Device Conditions Using a Violin Plot: The violin plot illustrates the distribution of crash types across various device conditions, providing insights into how traffic control devices impact the severity of crashes. Functioning properly devices have the most variation in crash types, with a significant proportion resulting in injury and/or tow due to crash, suggesting that even when devices are operational, other factors like human error or road conditions contribute to severe crashes. This highlights the need for further investigation into human behavior, even in areas with functioning traffic control systems. On the other hand, device conditions such as functioning improperly or no controls show a relatively lower variance in crash types, but with more extreme outcomes leaning towards severe crashes. This could imply that malfunctioning or absent traffic devices can lead to more severe accidents, emphasizing the importance of maintaining and installing adequate traffic control systems to mitigate crash severity. The presence of worn reflective material and missing devices further supports the need for effective traffic control maintenance to prevent more severe outcomes in accidents.

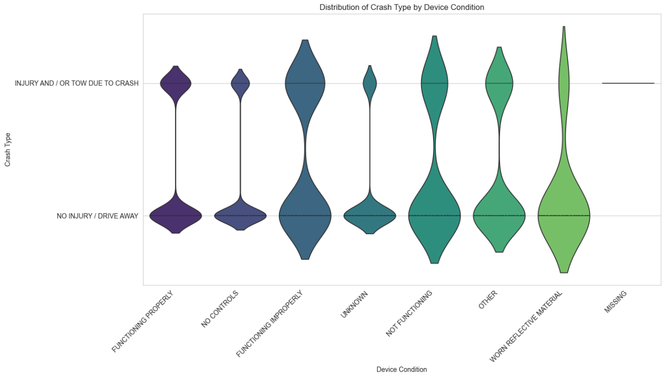

Fig. 8. Distribution of Crash Type by Device Condition

8) Distribution of Traffic Control Devices in Traffic Accidents: The pie chart displaying the distribution of traffic control devices involved in accidents provides a clear breakdown of how different traffic controls are linked to crash occurrences. The most significant portion of crashes involves no controls, indicating that areas without traffic regulation devices are particularly prone to accidents. This finding suggests a need for installing more traffic signals or control mechanisms in such locations to mitigate accident risks. Another notable observation is that a large proportion of accidents occur at traffic signals and stop signs, which may indicate issues related to driver behavior, such as ignoring signals or misjudging their timing. This emphasizes the importance of enforcing traffic rules more strictly at these intersections and potentially re-evaluating signal timings to ensure they are optimally configured for safety. Additionally, the relatively smaller segments involving devices like yield signs and pedestrian crossing signs highlight specific areas where further safety measures could be introduced to enhance protection for vulnerable road users like pedestrians.

9) Distribution of Damage Categories Across Different First Crash Types: The stacked bar plot depicting the damage categories by first crash type reveals key insights into the

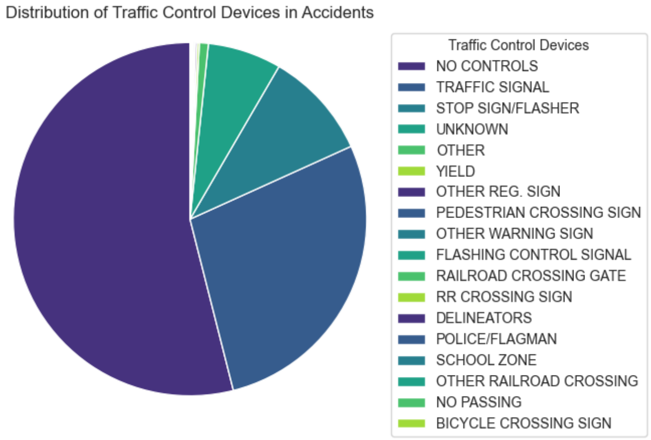

Fig. 9. Distribution of Traffic Control Devices in Accidents

extent of financial damage across different types of crashes. For most crash types, particularly angle collisions and rear-end collisions, the majority of crashes result in damages exceeding $1,500. This suggests that these types of accidents are likely to involve significant property damage, possibly due to the nature of the collisions where vehicles impact each other directly and with force.

In contrast, less severe collisions, such as pedestrian-related and fixed object crashes, display a more even distribution across the three damage categories, with some resulting in minimal damage (under $500). Non-collision incidents, including crashes with animals or other non-vehicle objects, tend to fall under the lower damage categories, which could indicate less severe impacts and minor vehicle repairs. These insights suggest that angle and rear-end collisions require more attention from insurance providers and vehicle safety experts, as they often result in costly repairs.

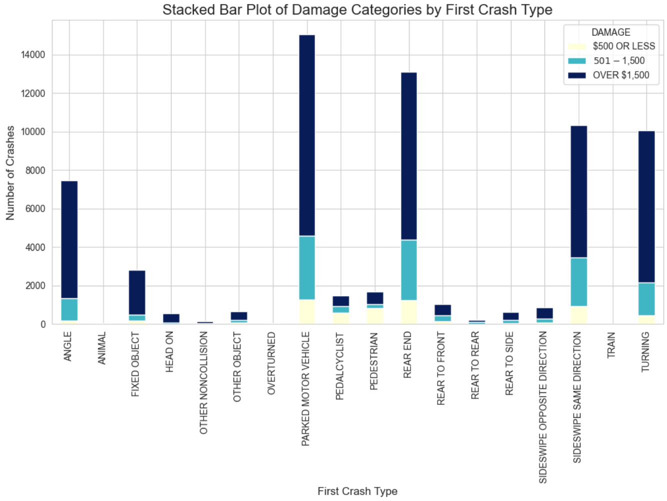

Fig. 10. Stacked Bar plot of Damage Categories by First Crash Type

10) Heatmap of Crashes by Road Conditions: The heatmap showing accidents by roadway alignment, surface condition, and road defects highlights several important patterns. The majority of accidents occurred on straight and level roads with dry surfaces (40,027 accidents), suggesting that even under favorable conditions, human factors like speeding, distraction, or following too closely may play a significant role in accidents. This finding underscores the need for continuous driver awareness and safety campaigns, even on seemingly safe roads. Interestingly, wet conditions on straight and level roads also show a relatively high number of accidents (6,394 incidents), indicating that adverse weather, such as rain, greatly increases the risk of accidents. Moreover, road defects such as debris on the roadway and worn surfaces appear to exacerbate these risks, particularly on straight and level roads, as evidenced by the significant accident counts in these categories. These findings suggest that maintaining road surfaces and clearing debris can have a substantial impact on reducing accident occurrences, particularly during adverse weather conditions.

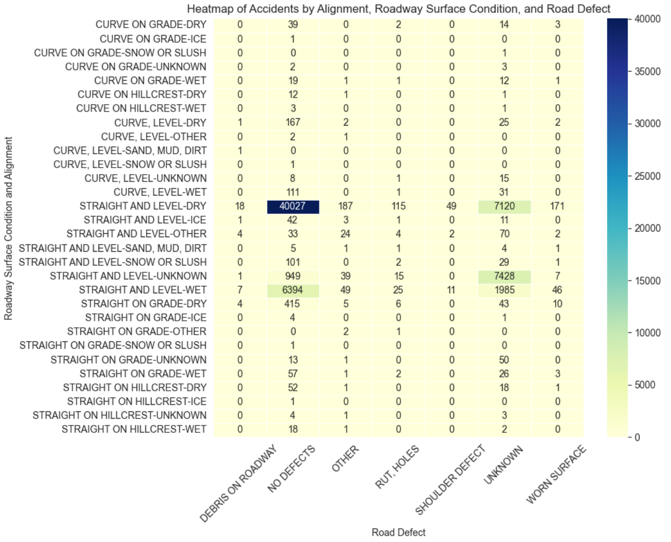

Fig. 11. Heatmap of Crashes by Alignment, Road way surface condition and Road defect

11) Distribution of Most Severe Injuries by First Crash Type: The bar chart illustrating the distribution of the most severe injuries by first crash type reveals several key insights. The majority of accidents, particularly those involving rear-end and angle collisions, tend to result in non-incapacitating injuries or no indication of injury, which implies that while these types of crashes are common, they are less likely to lead to severe or fatal injuries. This suggests that rear-end and angle collisions may occur more frequently due to traffic congestion or tailgating but tend to happen at lower speeds, reducing the severity of injuries. On the other hand, more severe outcomes, such as incapacitating injuries and fatalities, are associated with less frequent crash types, such as pedestrian collisions and fixed object impacts. These crash types, while less common, tend to be more dangerous, likely due to higher speeds or the vulnerability of pedestrians. The data underscores the need for targeted safety measures, such as pedestrian crossings and better vehicle control near high-risk zones like intersections, to reduce the likelihood of severe or fatal injuries in these types of accidents.

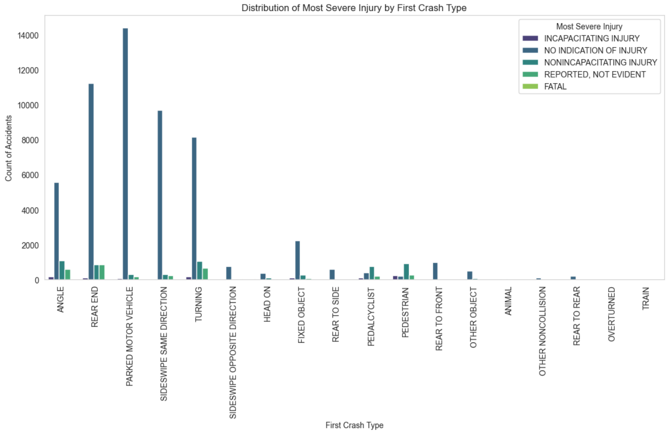

Fig. 12. Distribution of Most Severe Injuries by First Crash Type

12) Correlation Matrix of Traffic Accident Variables: The correlation matrix of traffic accident variables provides key insights into how different factors relate to one another. Notably, there is a strong positive correlation between the total number of injuries and the number of severe injuries (0.74), indicating that accidents involving a high number of injuries tend to also have more severe injuries. Additionally, the correlation between most severe injury and crash type (0.61) suggests that the nature of the crash plays a significant role in determining the injury severity. Another important observation is the negative correlation between posted speed limit and various injury variables. This indicates that higher posted speed limits may not directly correlate with more injuries, possibly due to higher speed limits being implemented in less congested areas with lower crash risk. Finally, the strong negative correlation between latitude and longitude (-0.98) is expected, as these are geographical variables that naturally move in opposite directions. These correlations can help in further refining traffic safety measures by focusing on the factors that most strongly influence crash severity and injury outcomes

13) Analysis of Average Reporting Delay by First Crash Type: The bar chart showing the average date difference by first crash type highlights several interesting trends. Crashes involving non-collision events, such as collisions with animals or other objects, exhibit the longest delay between the crash and police notification, with an average date difference of over 5 days. This could indicate that non-collision events are often considered less urgent or are less likely to result in immediate reporting, potentially due to lower perceived severity or logistical challenges in rural or remote areas. In contrast, crashes involving more severe or direct interactions, such as train accidents, turning collisions, and angle crashes, show a much shorter average date difference, often within a day or less. These crash types likely demand quicker intervention due to the higher risk of severe injuries and property damage. These insights suggest a potential gap in timely reporting for less severe or non-collision accidents, which could hinder data collection accuracy and delay appropriate responses in certain cases.

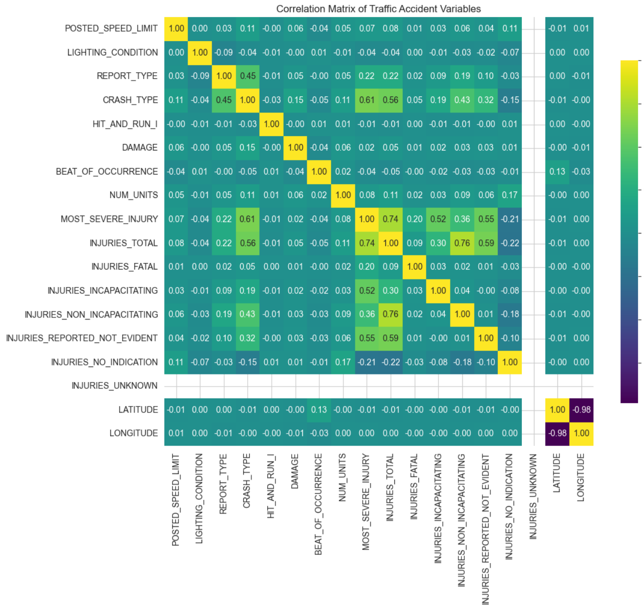

Fig. 13. Correlation Matrix of Traffic Accident Variables

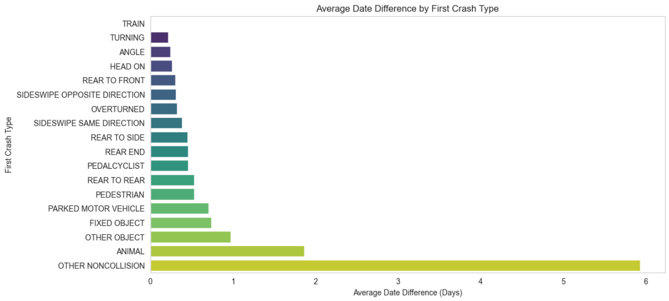

Fig. 14. Average Date Difference by First Crash type

14) Top 10 Streets with the Highest Number of Traffic Accidents in Chicago: The bar chart illustrating the 10 most dangerous streets in Chicago, based on the number of traffic accidents, reveals that Pulaski Road (1,806 accidents) and Western Avenue (1,747 accidents) are the two streets with the highest number of crashes. These findings suggest that these roads may have higher traffic volumes or potentially problematic road conditions, making them prone to accidents. Pulaski Road, in particular, stands out as the most accident-prone, indicating a need for targeted safety interventions such as traffic calming measures or better traffic signal management.

Additionally, other streets like Cicero Avenue (1,520 accidents) and Ashland Avenue (1,401 accidents) also rank high on the list, highlighting the importance of traffic safety improvements in these areas. These streets are known for heavy traffic flow, potentially explaining the elevated accident counts. The analysis underscores the need for authorities to focus on these high-risk roads to reduce traffic accidents and improve overall road safety in Chicago.

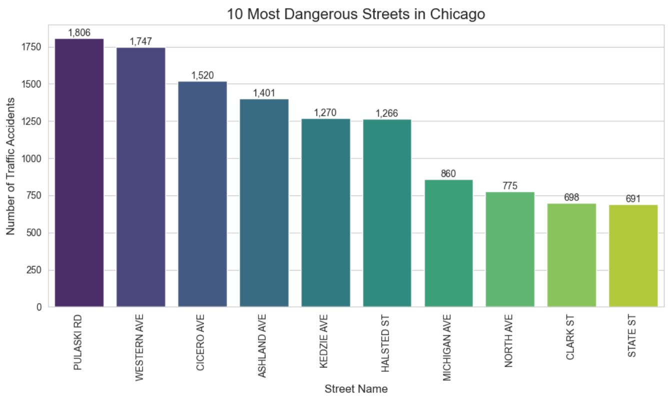

Fig. 15. 10 Most Dangerous Streets in Chicago

VI. ML Modeling and Algorithm Applications

The application of several machine learning algorithms was crucial to addressing key aspects of the traffic crash data analysis, including the prediction of crash types and the identification of high-risk factors contributing to severe accidents. Each algorithm was selected based on its suitability for classification tasks and its capacity to process the structured data provided in the Chicago Traffic Crashes dataset. The models applied include Logistic Regression, k-Nearest Neighbors (k-NN), Naive Bayes, Support Vector Machines (SVM), Random Forest Classifier and XGBoost Classifier all aimed at generating predictive insights from the data. Appropriate hyperparameters were tuned for each algorithm to optimize performance and ensure robust outcomes. The effectiveness of these models was evaluated using key performance metrics such as accuracy, precision, recall, and F1-score. Visualizations were also generated to aid in the interpretation of results, providing clarity on the factors influencing crash outcomes and damage levels. Overall, the integration of these machine learning techniques offers a data-driven approach to understanding urban traffic patterns, with the potential to inform strategies for improving road safety.

1) Models Tuning: The primary objective of this step was to identify the optimal hyperparameters for each machine learning model using the validation data, which would then be applied to the training dataset for improved model performance. Initially, the dataset was preprocessed by dropping categorical columns that were not required for the model training. Subsequently, the data was split into training, validation, and test sets, ensuring stratification to preserve class distributions. The features were scaled using StandardScaler to standardize the data, which is essential for algorithms like logistic regression, k-NN, and SVM that are sensitive to feature scaling.

Hyperparameter tuning was carried out using GridSearchCV for each model, which included Logistic Regression, k-NN, SVM, Random Forest, and XGBoost. A predefined set of hyperparameters was evaluated through cross-validation on the training data, and the best parameters for each model were selected based on the validation accuracy. This process ensured that the models were trained with the most optimal configurations, contributing to the overall improvement in predictive performance. The validation scores for the models ranged from 86% for k-NN to 89% for XGBoost, indicating the models' robust ability to generalize across unseen data.

2) Naive Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the features are independent given the class label, which is often referred to as the ”naive” assumption. Despite this strong assumption, Naive Bayes is highly effective for certain types of classification problems, particularly when the features are mostly independent. It is computationally efficient and performs well on large datasets, making it a good choice for a baseline model.

a) Model Tuning and Training: The Naive Bayes model was initialized using GaussianNB, which is suited for continuous input features that follow a Gaussian distribution. Since Naive Bayes doesn't require hyperparameter tuning, the model was directly trained on the scaled training data (X_train_scaled). The training was quick due to the simplicity of the model, making it highly efficient for this task.

b) Effectiveness and Metrics: The Naive Bayes model achieved a test accuracy of 85.6%, making it a decent baseline model for this classification problem. The classification report provides further insights:

- Class 0 (Negative Class) was handled effectively with a precision of 85% and a recall of 97%, showing that the model excelled at identifying and correctly classifying negative instances. The F1-score of 91% reflects a strong balance between precision and recall for the negative class.
- Class 1 (Positive Class) demonstrated a precision of 90%, meaning the model was good at identifying positive cases. However, the recall was lower at 57%, indicating that it missed a substantial number of positive instances. The F1-score of 70% highlights this trade-off between precision and recall, where the model is slightly less effective at capturing all positive cases.

c) Insights Gained: Naive Bayes, while simple, provides a quick and effective baseline model for binary classification tasks. Its major advantage is computational efficiency, but its performance can be limited in cases where the independence assumption doesn't hold true. The lower recall for Class 1 highlights that this model may not be the best for problems where identifying positive cases is crucial.

d) Confusion Matrix Insights: The confusion matrix provides a clear view of where the Naive Bayes model performs well and where it struggles:

- True Negatives (9128): The model correctly predicted a large number of negative cases, showing strong performance for the majority class.

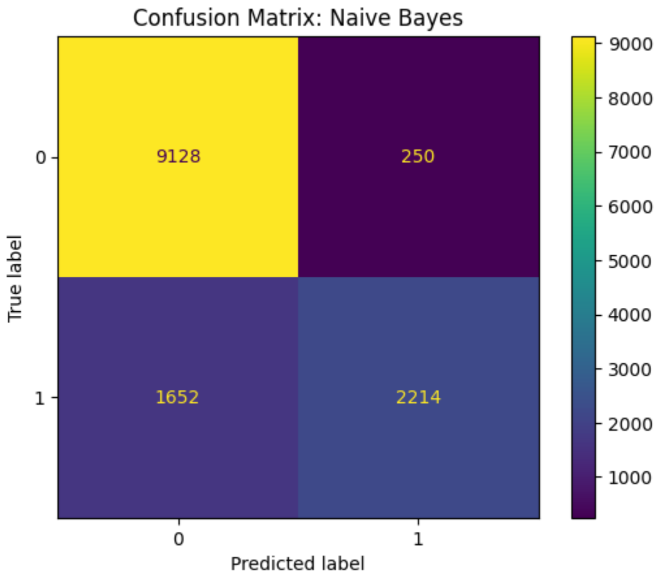

Fig. 16. Confusion Matrix: Naive Bayes

- True Positives (2214): The model successfully identified a good number of positive cases but missed some, reflected in the false negative count.
- False Positives (250): There are relatively few false positives, indicating that when the model predicts a positive case, it is usually correct.
- False Negatives (1652): The model struggles more with false negatives, meaning it missed several positive cases, contributing to the lower recall for Class 1.

e) Conclusion: Naive Bayes achieved a test accuracy of 85.6%, making it a reasonable starting point for this problem. While the model performs well in identifying negative cases, its recall for positive cases could be improved. Naive Bayes is useful as a quick and efficient model but may need to be replaced or supplemented by more complex models if the recall for positive cases is critical for the application.

3) Logistic Regression: Logistic Regression was chosen because it is a straightforward, interpretable model that works well for binary classification problems. It's particularly effective when the relationship between features and the target variable is linear, and it can provide probabilities for class membership, which is useful for understanding the confidence of the model's predictions. Given the need for a baseline model that can offer insights into the data and provide solid classification performance, Logistic Regression was a fitting choice.

a) Model Tuning and Training: We used GridSearchCV to perform hyperparameter tuning. The key hyperparameters we tuned were:

- Regularization strength (C): This controls the trade-off between bias and variance, ensuring the model generalizes well without overfitting.
- Solver: We explored solvers like liblinear and lbfgs to optimize model training.

Cross-validation (3-fold) was used to select the best model based on validation accuracy, after which the selected Logistic Regression model was trained on the scaled training data (X_train_scaled). It was then tested on the test set (X_test_scaled).

b) Effectiveness and Metrics: Logistic Regression achieved a test accuracy of 89.1%, reflecting its ability to classify the majority of cases correctly. Precision, recall, and F1-score provide more detailed insights:

- Class 0 (Negative Class) was handled effectively with a precision of 89% and recall of 96%, meaning the model was highly effective at identifying and correctly classifying negative instances. The high recall indicates that the model captured nearly all of the negative cases, and the F1-score of 93% reflects a balance between precision and recall.
- Class 1 (Positive Class) showed strong precision of 88%, but the recall was lower at 73%, indicating that while the model was generally good at identifying positive cases, it missed a notable proportion of them. The F1-score of 79% reflects this trade-off between precision and recall for the positive class.

The overall weighted average F1-score of 89% indicates that Logistic Regression maintains consistent performance across both classes, with a strong balance between precision and recall.

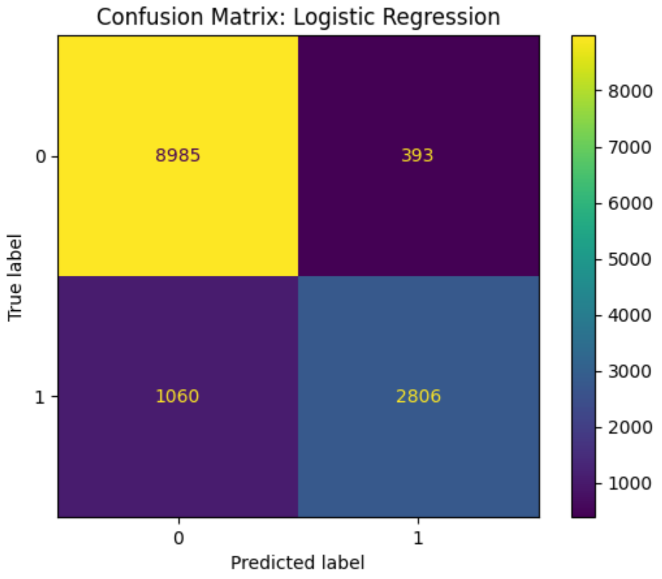

Fig. 17. Confusion Matrix: Logistic Regression

c) Confusion Matrix Insights: The confusion matrix helps visualize where the model excels and where it struggles:

- True Negatives (8985): These represent correctly predicted negative cases, which is a strong point of the model.

- True Positives (2806): These are the correctly predicted positive cases, which show the model's effectiveness at identifying positive instances.
- False Positives (393): These are cases where the model incorrectly predicted positive outcomes, suggesting a slight over-prediction of the positive class, though the number is relatively low.
- False Negatives (1060): These are the missed positive cases, where the model predicted negative when the actual class was positive. This higher number of false negatives is reflected in the lower recall for Class 1, meaning that the model may not be capturing all positive cases as well as it could.

d) Conclusion and Insights: Logistic Regression provided a solid performance, with an accuracy of 89.1%, and demonstrated strengths in identifying negative cases. However, the recall for positive cases (Class 1) could be improved. The model's interpretability also allowed us to identify which features were contributing most to predictions, offering useful business insights.

4) K-Nearest Neighbors (KNN): K-Nearest Neighbors (KNN) is a simple yet powerful non-parametric algorithm that classifies data points based on the majority class of their nearest neighbors. We selected KNN for this problem because it does not assume any underlying data distribution and performs well in scenarios where local patterns in the data are important. Additionally, KNN provides an alternative perspective to linear models like Logistic Regression by focusing on distance-based classification. This allows us to see how well the data can be classified based on its neighbors.

a) Model Tuning and Training: We used GridSearchCV to tune the following key hyperparameters:

- Number of neighbors (n_neighbors): We explored values like 3, 5, and 7 to determine the optimal number of neighbors for classification.
- Weights: We compared uniform (all neighbors have equal weight) versus distance (closer neighbors are weighted more heavily).
- Distance metric (p): Manhattan (p=1) and Euclidean (p=2) distances were tested to determine the best distance calculation method for this dataset.

After hyperparameter tuning, we trained the best KNN model on the scaled training data (X_train_scaled) and validated it on the scaled test data (X_test_scaled). The results showed that KNN performed better than Naive Bayes, highlighting its ability to capture patterns in the dataset more effectively when relying on local neighbor information.

b) Effectiveness and Metrics: The KNN model achieved a test accuracy of 85.8%, which is higher than Naive Bayes but still lower than Logistic Regression and other models like Random Forest and XGBoost. KNN demonstrated solid performance in classifying the majority class but showed limitations in detecting the minority class (Class 1).

- Class 0 (Negative Class): The model performed well with precision of 86% and recall of 96%, meaning that KNN accurately classified most negative instances. The F1-score for Class 0 is 90%, indicating strong performance in identifying and correctly predicting negative cases.
- Class 1 (Positive Class): The precision for Class 1 was 85%, but the recall dropped to 62%, indicating that the model missed a significant portion of positive cases. This is reflected in the F1-score of 72%, showing that KNN struggled to identify all positive instances, leading to a higher number of false negatives.

The macro average F1-score of 81% and weighted average F1-score of 85% show that while KNN is effective at predicting negative cases, it faces challenges in correctly identifying all positive cases.

c) Insights Gained: KNN showed that local patterns in the dataset are important, and it was effective for the majority class. However, the model's tendency to misclassify some positive cases suggests that it might require further fine-tuning or alternative methods to handle class imbalance better. The performance could be improved by adjusting hyperparameters, using a larger number of neighbors, or considering weighted KNN to better capture the minority class.

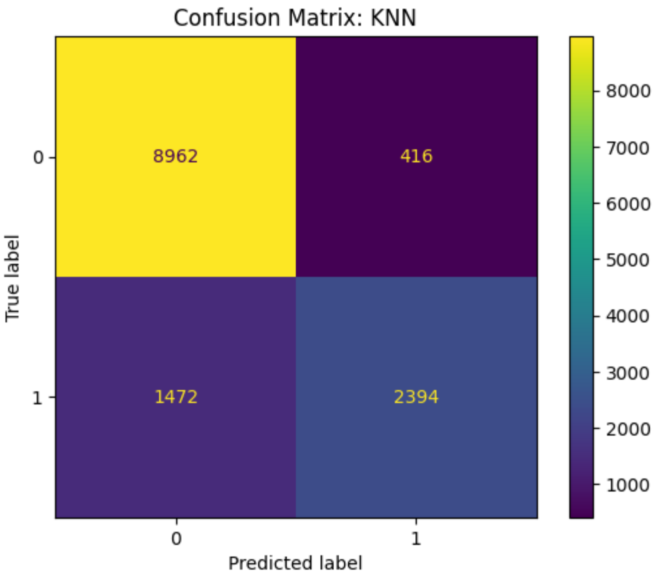

Fig. 18. Confusion Matrix: KNN

d) Confusion Matrix Insights: The confusion matrix for KNN reveals the following details:

- True Negatives (8962): The model correctly predicted 8962 instances of Class 0, showing strong performance in detecting negative cases.
- True Positives (2394): KNN correctly identified 2394 instances of Class 1, which is an improvement over Naive Bayes but still leaves room for more accurate detection of positive cases.
- False Positives (416): The model incorrectly predicted 416 instances as positive when they were actually negative, showing a relatively low false positive rate.

- False Negatives (1472): KNN missed 1472 instances of Class 1, leading to a lower recall for the positive class and highlighting an area for improvement.

e) Conclusion: With a test accuracy of 85.8%, KNN outperformed Naive Bayes and showed reasonable classification capability for the majority class. However, like Naive Bayes, it struggled with identifying positive cases, as indicated by the higher number of false negatives. In comparison to Naive Bayes, KNN provided more accurate predictions but still requires refinement, particularly in improving recall for Class 1.

5) Support Vector Machine (SVM): Support Vector Machines (SVM) are powerful classifiers that work well in both linear and non-linear classification tasks. SVM was chosen because it can handle high-dimensional data effectively and provides a robust approach to separating classes by maximizing the margin between them. This ability to find the optimal hyperplane for classification makes SVM a strong candidate for this problem, especially when the dataset may not be perfectly linearly separable. SVM also works well in scenarios with complex decision boundaries, which could improve the model's performance in distinguishing between classes.

a) Model Tuning and Training: Using GridSearchCV, we tuned the SVM hyperparameters to find the best configuration for the dataset:

- Regularization parameter (C): We tested different values of C to control the trade-off between maximizing the margin and minimizing classification errors.
- We used a linear kernel for simplicity and computational efficiency, but SVM can also use non-linear kernels such as polynomial or RBF for more complex datasets.

Once the best parameters were identified, the SVM model was trained on the scaled training data (X_train_scaled) and validated on the test set (X_test_scaled).

b) Effectiveness and Metrics: SVM achieved a test accuracy of 88.3%, showing strong classification performance, slightly better than KNN and Naive Bayes but slightly lower than Logistic Regression. SVM performed well on both the majority class (Class 0) and the minority class (Class 1), but it had some difficulty in correctly identifying all positive instances.

- Class 0 (Negative Class): SVM provided high precision of 89% and recall of 95%, meaning it correctly identified most negative cases. The F1-score of 92% for Class 0 highlights its effectiveness in accurately classifying the majority class.
- Class 1 (Positive Class): For the positive class, SVM demonstrated precision of 87% and a recall of 71%, which is better than models like Naive Bayes and KNN. However, the recall indicates that there is still room for improvement in minimizing false negatives. The F1-score for Class 1 is 78%, reflecting the model's performance in identifying positive cases.

The macro average F1-score of 85% and weighted average F1-score of 88% show that SVM offers a balanced performance across both classes, though improvements in recall for Class 1 could enhance the model's effectiveness.

c) Insights Gained: SVM is particularly effective when the data is not linearly separable. Its ability to optimize the decision boundary helps it outperform simpler models like Naive Bayes and KNN. However, SVM's performance may vary depending on the selected regularization parameter, and it can be computationally expensive for large datasets. Despite its good performance, improving recall for the minority class (Class 1) may be a priority, especially if missing positive cases is costly.

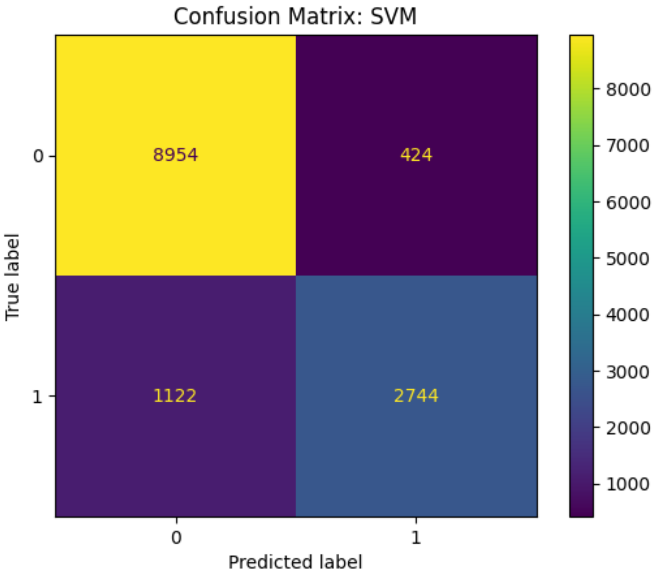

Fig. 19. Confusion Matrix: SVM

d) Confusion Matrix Insights: The confusion matrix for SVM reveals the following details:

- True Negatives (8954): SVM correctly identified 8954 instances of Class 0, showing that it handles the majority class effectively.
- True Positives (2744): The model successfully predicted 2744 instances of Class 1, which is an improvement over models like Naive Bayes and KNN, indicating that SVM captures more positive cases.
- False Positives (424): The model incorrectly predicted 424 instances as positive when they were actually negative, reflecting a slight overestimation of the positive class.
- False Negatives (1122): SVM missed 1122 instances of Class 1, which is lower than the number of false negatives in KNN and Naive Bayes, suggesting improved recall for Class 1.

e) Conclusion: With a test accuracy of 88.3%, SVM demonstrated strong performance, particularly in improving recall for the positive class (Class 1) compared to previous models like KNN and Naive Bayes. While it performs well in both precision and recall, SVM's overall performance could still be enhanced by further tuning, particularly to minimize false negatives in sensitive applications. Nonetheless, its ability to find optimal margins for classification makes SVM a powerful and reliable model for this problem.

6) Random Forest: Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It was chosen for this problem because it typically performs well on classification tasks and can handle large datasets with many features. Random Forest's ability to reduce variance and improve prediction stability makes it a strong candidate for this classification task. Additionally, it handles feature importance naturally, which provides insights into which features contribute most to the predictions.

a) Model Tuning and Training: We used GridSearchCV to tune key hyperparameters for Random Forest:

- Number of trees (n_estimators): Different values such as 50, 100, and 200 were tested to find the optimal number of trees.
- Maximum depth (max_depth): This controls how deep the trees can grow, preventing overfitting by limiting depth.
- Minimum samples split (min_samples_split): This parameter was tuned to determine the minimum number of samples required to split a node.

After tuning, the best Random Forest model was trained on the scaled training data (X_train_scaled) and evaluated on the test data (X_test_scaled).

b) Effectiveness and Metrics: Random Forest achieved a test accuracy of 89.2%, which is comparable to Logistic Regression and SVM. The model performed well across both classes, but similar to other models, it showed some difficulty in capturing all positive instances (Class 1), as reflected in the confusion matrix.

- Class 0 (Negative Class): The model performed very well for Class 0, with precision of 89% and recall of 97%, indicating that it correctly identified the vast majority of negative cases. The F1-score of 93% reflects the model's effectiveness in classifying negative instances.
- Class 1 (Positive Class): For Class 1, the model exhibited precision of 91%, showing that it accurately predicted positive cases. However, the recall for Class 1 was 70%, meaning that some positive cases were missed. The F1-score for Class 1 is 79%, demonstrating an overall strong performance for the positive class, better than models like KNN and Naive Bayes.

The macro average F1-score of 86% and weighted average F1-score of 89% indicate that Random Forest maintains balanced performance across both classes, with slightly higher precision for Class 1, though recall for the positive class could still be improved.

c) Insights Gained: Random Forest is a powerful model for classification tasks, and its ensemble nature makes it less prone to overfitting compared to individual decision trees. The model's feature importance capabilities allow us to gain insights into which features are most influential in classification. The high accuracy and precision for both classes indicate that the model provides stable and reliable predictions. The lower recall for Class 1 suggests that further improvements could be made to increase sensitivity to positive cases, especially if missing positive instances is costly.

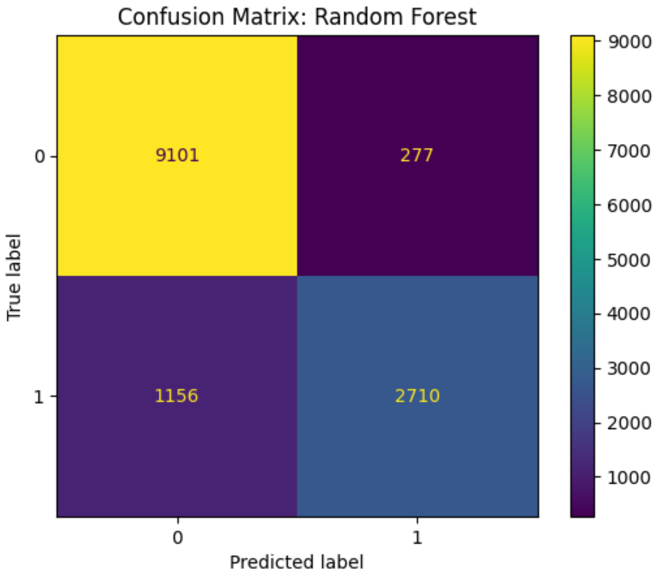

Fig. 20. Confusion Matrix: Random Forest

d) Confusion Matrix Insights: The confusion matrix for Random Forest provides the following insights:

- True Negatives (9101): Random Forest correctly identified 9101 instances of Class 0, indicating that it is highly effective at detecting the negative class.
- True Positives (2710): The model successfully predicted 2710 instances of Class 1, showing strong performance in detecting positive cases, better than models like Naive Bayes and KNN.
- False Positives (277): The model predicted 277 instances as positive when they were actually negative, which is a relatively low number of false positives.
- False Negatives (1156): Random Forest missed 1156 instances of Class 1, indicating that while it captures a large portion of positive cases, there is room for improving recall.

e) Conclusion: With a test accuracy of 89.2%, Random Forest performed very well, handling both classes effectively. The model's strength lies in its ability to handle large datasets with multiple features and reduce overfitting through ensemble learning. Its performance was slightly better than SVM and on par with Logistic Regression, making it a strong model for this classification problem. However, as with other models, further improvement in recall for the positive class could be beneficial if identifying positive instances is a priority.

7) XGBoost (XGB): XGBoost (Extreme Gradient Boosting) is a powerful ensemble learning algorithm that builds multiple decision trees in sequence to minimize the error in predictions. It was chosen for this problem due to its high efficiency, ability to handle large datasets, and robustness in reducing both bias and variance. XGBoost also has built-in regularization to prevent overfitting, making it a strong choice for complex classification tasks. It generally performs well in structured datasets with a mix of continuous and categorical features.

a) Model Tuning and Training: For XGBoost, we used GridSearchCV to tune hyperparameters such as:

- Number of estimators (n_estimators): We tested values like 50, 100, and 200 to determine the optimal number of trees.
- Learning rate (learning_rate): Different values like 0.01, 0.1, and 0.2 were tested to control the contribution of each tree.
- Maximum depth (max_depth): We experimented with different depths (3, 6, and 9) to control the complexity of each tree.

After hyperparameter tuning, the best XGBoost model was trained on the scaled training data (X_train_scaled) and tested on the scaled test data (X_test_scaled).

b) Effectiveness and Metrics: XGBoost achieved a test accuracy of 89.3%, which is slightly better than Random Forest and comparable to Logistic Regression. XGBoost was effective in handling both classes but, like other models, still showed some difficulty in fully capturing positive instances (Class 1). However, its ability to perform well on both classes makes it a strong candidate for this classification task.

- Class 0 (Negative Class): XGBoost performed exceptionally well, with precision of 89% and recall of 96%, meaning it correctly classified the vast majority of negative cases. The F1-score of 93% highlights its strong performance for Class 0.
- Class 1 (Positive Class): The model exhibited precision of 89% for Class 1, showing that it was effective in predicting positive cases. The recall for Class 1 was 73%, meaning that while the model captured most positive instances, it missed a significant portion. The F1-score for Class 1 was 80%, reflecting its overall better performance compared to models like KNN and Naive Bayes, though there is still room for improvement in reducing false negatives.

The macro average F1-score of 86% and weighted average F1-score of 89% demonstrate that XGBoost delivers strong and balanced classification across both classes, making it one of the most effective models in the analysis.

c) Insights Gained: XGBoost provides valuable insights by offering feature importance scores, which help identify which features are most relevant to the classification task. Its high performance in both precision and recall makes it a solid choice for this dataset. The model's ability to handle complex patterns in the data through boosting improves overall prediction accuracy. However, as with other models, recall for the positive class could be further improved to reduce the number of missed positive cases.

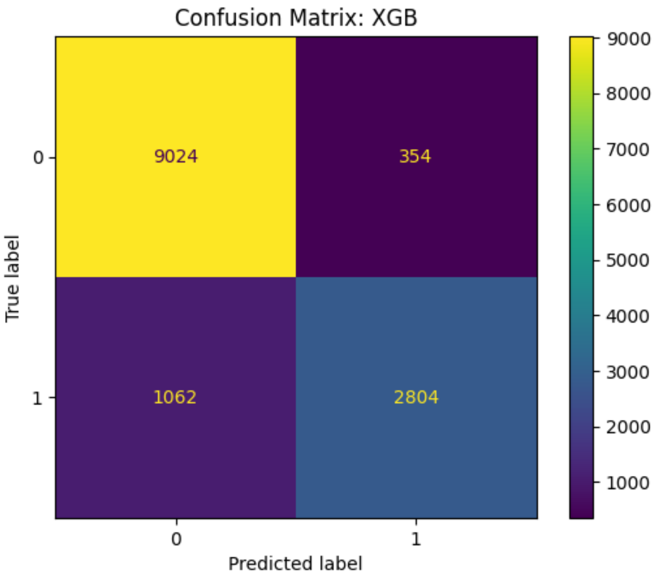

Fig. 21. Confusion Matrix: XGBoost

d) Confusion Matrix Insights: The confusion matrix for XGBoost shows the following insights:

- True Negatives (9024): The model correctly identified 9024 instances of Class 0, indicating that it handles the majority class very well.
- True Positives (2804): XGBoost correctly identified 2804 instances of Class 1, showing strong performance in detecting positive cases and outperforming models like KNN and Naive Bayes.
- False Positives (354): The model predicted 354 instances as positive when they were actually negative, which is a low false positive rate, reflecting high precision.
- False Negatives (1062): XGBoost missed 1062 instances of Class 1, which is comparable to the performance of Random Forest, but still leaves room for improvement in detecting all positive cases.

e) Conclusion: With a test accuracy of 89.3%, XGBoost performed very well in classifying both the negative and positive classes, showing a balanced performance. The model's ability to handle complex patterns and its built-in regularization make it a powerful choice for this problem. While the overall performance is strong, as with other models, improving the recall for the positive class could be beneficial for applications where correctly identifying positive cases is critical. XGBoost's ability to provide feature importance also adds an extra layer of interpretability to the model's predictions.

VII. Comparative Model Evaluation and Visualization Analysis

We present and analyze different plots to compare the performance of the models, focusing on key evaluation metrics and insights derived from their application to our classification task. These include ROC curves, feature importance for Random Forest and XGBoost, log loss over time for both Logistic Regression and XGBoost, and an accuracy plot for Random Forest. These visualizations provide valuable insights into how each model performs, helping us understand their strengths and weaknesses in terms of classification accuracy and feature importance.

A. ROC Curves

ROC (Receiver Operating Characteristic) curves illustrate the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) at different threshold settings. The Area Under the Curve (AUC) is a summary measure of the model's performance across all thresholds, where a higher AUC indicates a better-performing model.

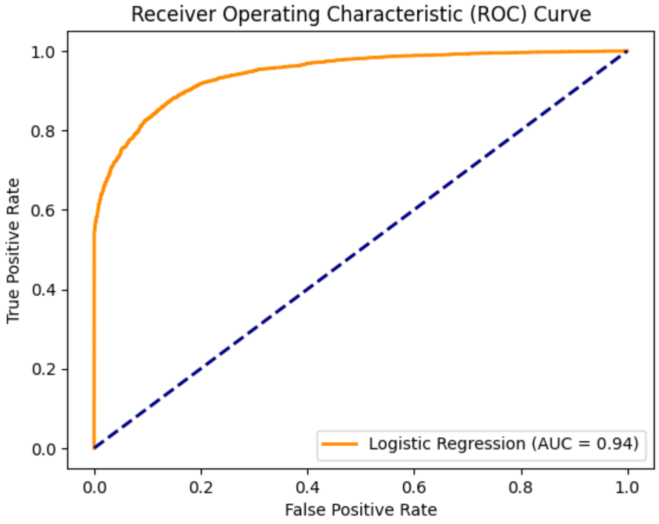

Fig. 22. ROC Curve: Logistic Regression

1) ROC Curve for Logistic Regression: The ROC curve for Logistic Regression shows a high AUC of 0.94, indicating strong model performance with a good balance between true positive and false positive rates. Logistic Regression is able to correctly classify a large proportion of positive cases while keeping false positives relatively low.

2) ROC Curve for Random Forest: Similarly, Random Forest achieved an AUC of 0.94, demonstrating that it is highly effective in distinguishing between the two classes. The model's ability to use multiple decision trees enhances its robustness and stability, leading to comparable performance to Logistic Regression.

3) ROC Curve for XGBoost: XGBoost performed the best, with an AUC of 0.95. This suggests that XGBoost is slightly better at classifying positive cases while maintaining a low false positive rate. The boosting technique allows XGBoost to minimize classification errors iteratively, resulting in superior performance compared to Logistic Regression and Random Forest.

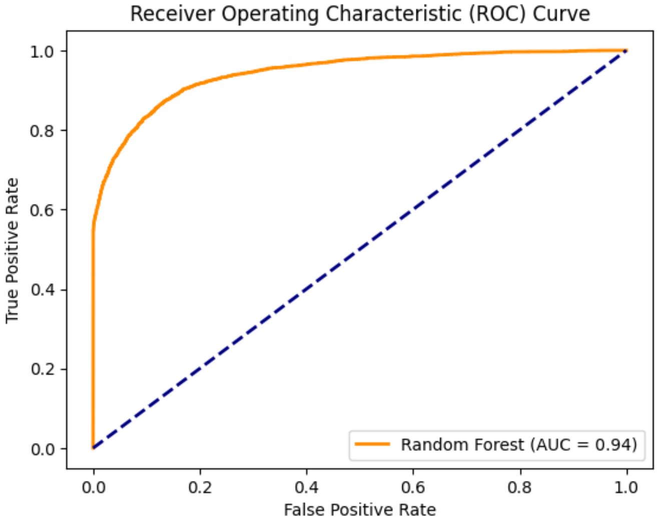

Fig. 23. ROC Curve: Random Forest

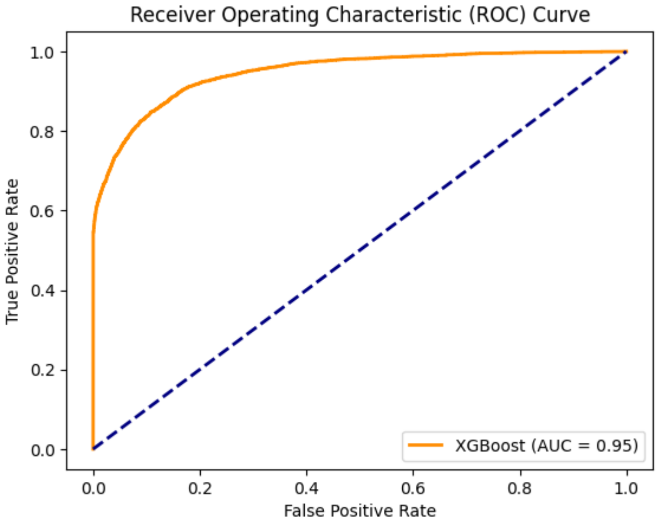

Fig. 24. ROC Curve: XGBoost

B. Feature Importance

Feature importance measures provide insights into which features contribute most to the model's predictions. Understanding feature importance can help prioritize variables for decision-making and feature engineering.

1) Feature Importance for Random Forest: The Random Forest model identified INJURIES_TOTAL and MOST_SEVERE_INJURY as the top two most important features, which makes sense in the context of classifying accident severity or outcomes. These features have the highest impact on the model's predictions, while other features like REPORT_TYPE and LOCATION also contribute, but to a lesser extent.

2) Feature Importance for XGBoost: XGBoost also highlighted MOST_SEVERE_INJURY as the most influential fea-

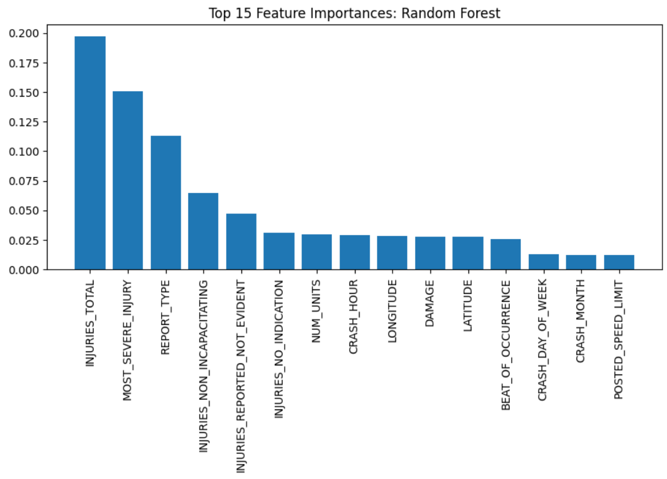

Fig. 25. Feature Importance: Random Forest

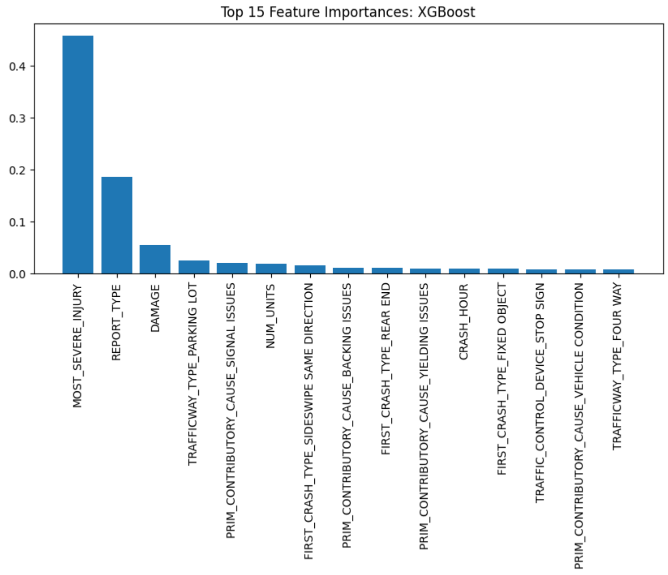

Fig. 26. Feature Importance: XGBoost

ture, followed by REPORT_TYPE and DAMAGE. The feature importance for XGBoost is more concentrated on a few key features, suggesting that the model relies heavily on specific variables for making accurate predictions.

C. Log Loss Over Time

Log loss measures the performance of a classification model where the output is a probability between 0 and 1. Lower log loss indicates better model performance. We tracked log loss over iterations for both Logistic Regression and XGBoost to understand how the models improved with training.

1) Log Loss for Logistic Regression: The log loss curve for Logistic Regression shows a rapid decrease in both train and test loss within the first 20 iterations, after which the loss plateaus. The train and test losses remain close, indicating that the model is well-fitted without overfitting, and it performs consistently across both datasets.

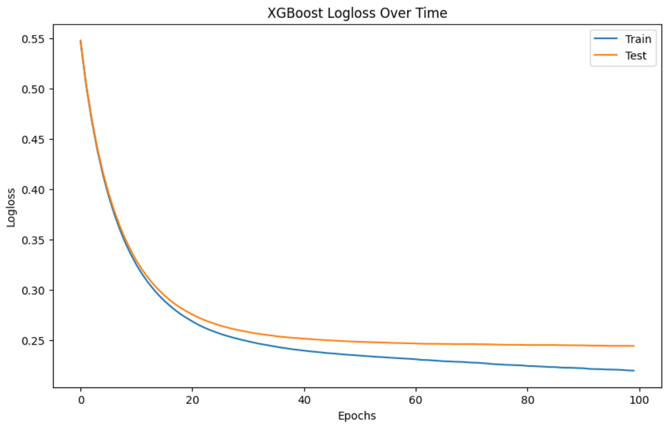

Fig. 27. Log Loss: Logistic Regression

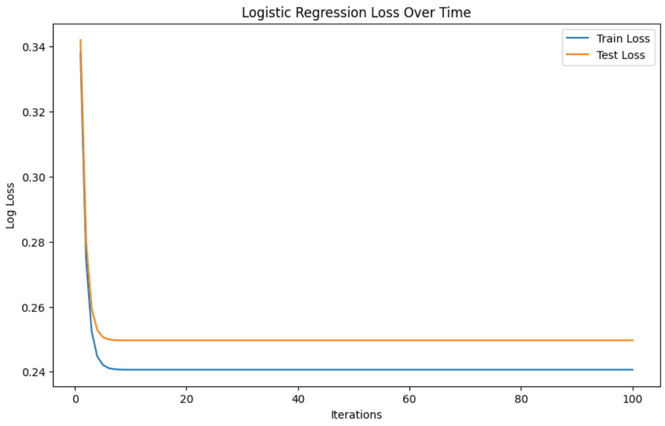

Fig. 28. Log Loss: XGBoost

2) Log Loss for XGBoost: XGBoost's log loss curve shows a steep decline in the initial iterations, with both train and test losses converging as training progresses. The train loss is slightly lower than the test loss, indicating that the model fits the training data slightly better, but the performance gap is minimal. XGBoost maintains low log loss over time, confirming its robustness.

D. Random Forest Accuracy Plot

The Random Forest Accuracy Plot tracks how both the training and test accuracy change as the number of trees (n_estimators) increases from 1 to 100. This helps us understand how Random Forest performs as more trees are added and when it begins to overfit the training data.

1) Train Accuracy (blue line): The train accuracy increases rapidly and approaches 100% as the number of trees increases, indicating that the model is overfitting the training data.

2) Test Accuracy (orange line): The test accuracy improves quickly within the first 20-30 trees and stabilizes around 88%, indicating that adding more trees does not lead to significant improvements beyond this point.

3) Insights from the Accuracy Plot: Optimal Number of Trees: Around 20-30 trees seem to provide stable performance

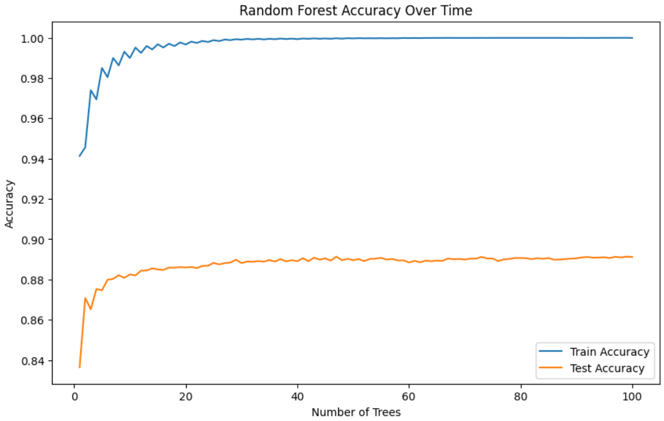

Fig. 29. Random Forest Accuracy Plot

on the test set, after which the gains in test accuracy become marginal.

4) Overfitting: The growing gap between train and test accuracy shows that Random Forest tends to overfit the training data when too many trees are added.

This plot helps guide decisions on the number of trees to include in the final model to prevent overfitting while maintaining strong generalization performance.

E. Insights from the Visualisations

Through the ROC curves, feature importance, log loss, and accuracy plots, we gained several insights into how each model performs:

XGBoost has the highest AUC and slightly better feature importance and log loss scores, making it the best performer overall.

Random Forest and Logistic Regression both showed strong performance with high AUC scores and reasonable feature importance distributions.

The feature importance analysis provided key insights into the most influential variables across models, particularly the importance of injury-related features in predicting accident outcomes.

The log loss and accuracy analysis for Random Forest demonstrated that the model achieves strong generalization performance with relatively few trees, though adding more trees leads to overfitting.

VIII. Conclusion

We applied and evaluated multiple machine learning models to solve our classification problem, including Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost. Each model was trained, tuned, and validated using several performance metrics, such as accuracy, precision, recall, ROC AUC, log loss, and feature importance.

After a thorough evaluation, XGBoost emerged as the most effective model, achieving the highest test accuracy of 89.3% and the best ROC AUC score of 0.95. XGBoost's ability to iteratively minimize classification errors and its fine-tuned hyperparameters made it excel at correctly classifying both the majority and minority classes. Additionally, XGBoost provided useful insights into feature importance, highlighting critical factors such as the severity of injuries and report types. These insights are valuable for understanding key factors influencing the classification outcomes.

Random Forest, while slightly less accurate than XGBoost, also performed well, with a test accuracy of 89.2% and an AUC of 0.94. However, Random Forest showed signs of overfitting as the number of trees increased, which was evident from the accuracy plot. Despite this, its feature importance analysis aligned well with XGBoost, indicating that both models captured similar influential features.

Models like Logistic Regression and SVM also performed admirably, each showing strong accuracy and AUC scores, but they were outperformed by the ensemble models, especially XGBoost. Naive Bayes and KNN, while useful as baselines, were less effective for this classification task, struggling particularly with minority class predictions.

In conclusion, XGBoost's high accuracy, strong AUC score, and robust feature importance analysis make it the optimal choice for this classification problem. Its ability to handle complex relationships in the data while maintaining low log loss across iterations positions it as the superior model for real-world application in this scenario

References

[1] U.S. Department of Transportation, "Traffic Crashes Dataset," data.gov, 2023. [Online]. Available: https://catalog.data.gov/dataset/traffic-crashes-crashes/resource/858674f2-8acc-4803-ba50-91c7faf54030. [Accessed: Month Day, Year].

[2] C. O'Neill and R. Schutt, *Doing Data Science*, O'Reilly, 2013.

[3] National Institute of Standards and Technology (NIST), "Exploratory Data Analysis," 2021. [Online]. Available: https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm. [Accessed: February 2021].

[4] John Tukey Biography, [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Tukey/. [Accessed: 2021].

[5] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: https://link.springer.com/article/10.1023/A:1010933404324.

[6] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in Proc. Springer Ensemble Mach. Learn., 2012, pp. 157–175, doi: https://link.springer.com/book/10.1007/978-1-4419-9326-7.

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 785-794. DOI: 10.1145/2939672.2939785.

[8] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, 2009, Art. no. 1883, doi: 10.4249/scholarpedia.1883 http://www.scholarpedia.org/article/K-nearest_neighbor.

[9] L. C. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in Proc. IEEE Int. Conf. Data Mining, 2002, pp. 306–313, doi: 10.1109/icdm.2002.1183917 https://ieeexplore.ieee.org/document/1183917.