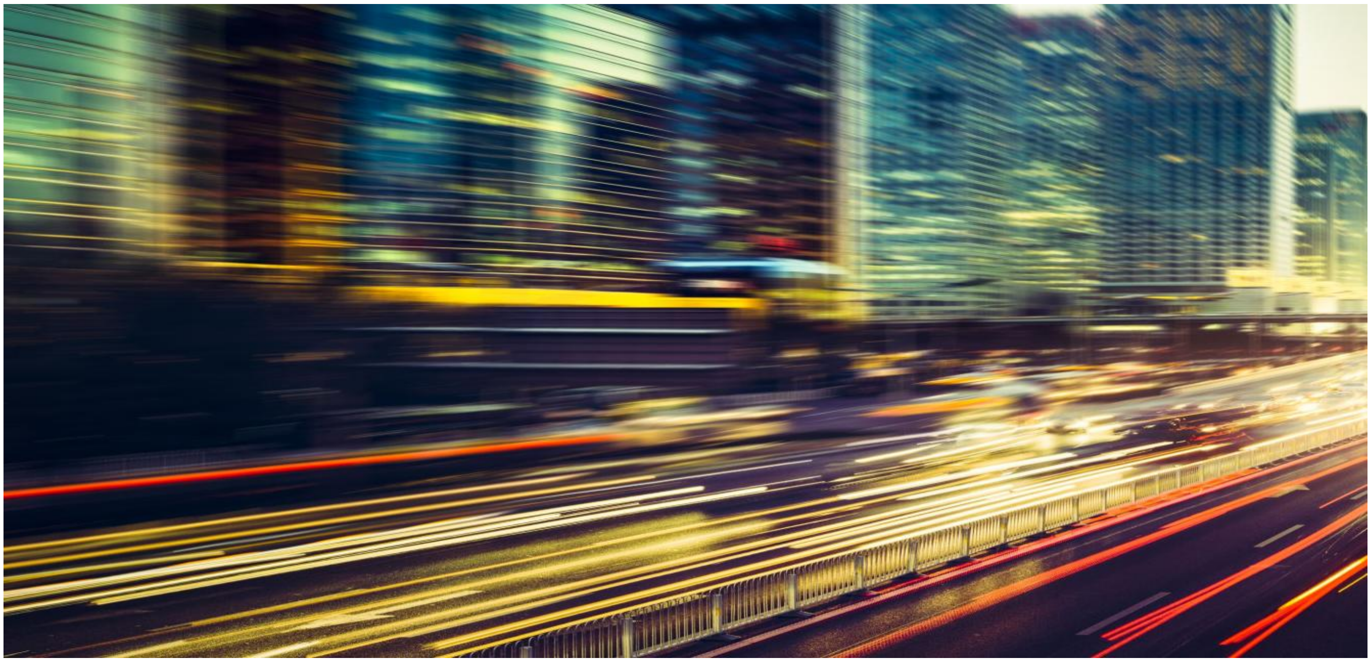
Detailed Analysis and Summary Report on Traffic Crash Analysis Project



Big Data - Fall 2024

Team Data Pioneers

- **Garima Dubey (U22534435)**
- **Janhavi Kharmale (U30004934)**
- **Ruthvik Bacha U12145480**
- **Raajitha Sai U82567912**

1. Project Overview

The project focuses on leveraging machine learning to predict the severity of traffic crashes using the Chicago Traffic Incident Dataset. The primary aim is to identify key contributing factors influencing crash severity and develop models that assist in proactive safety planning. This initiative is particularly valuable for city planners and policymakers to implement measures that reduce severe crashes and enhance road safety.

2. Data Preparation and Preprocessing

The dataset contained diverse attributes such as time, location, weather conditions, road characteristics, and outcomes of traffic incidents. Key data preparation steps included:

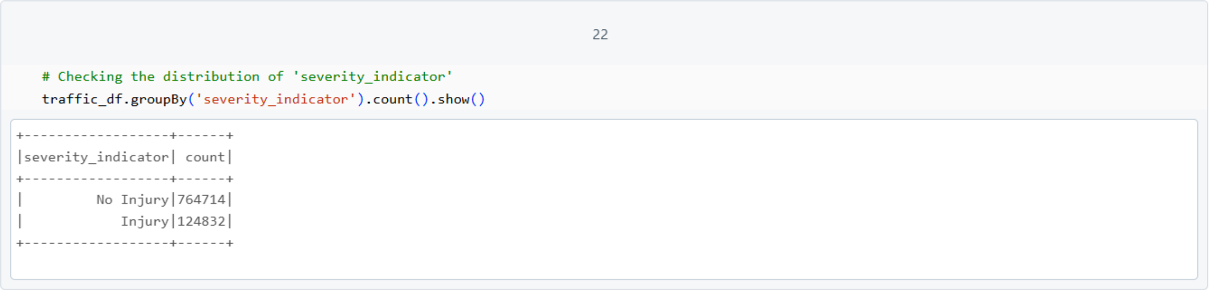
Data Cleaning: Addressed null values and inconsistencies by removing or imputing missing data. This step ensured the integrity of the dataset for training robust models. Approximately 10% of records had missing values, which were handled through targeted imputation and removal strategies.

Categorical Data Handling: Encoded categorical variables using techniques like one-hot encoding and label encoding to convert them into numerical formats suitable for machine learning algorithms.

Created additional features, such as time buckets (e.g., peak hours vs. non-peak hours) and weather condition flags, to enrich the dataset and improve model learning.

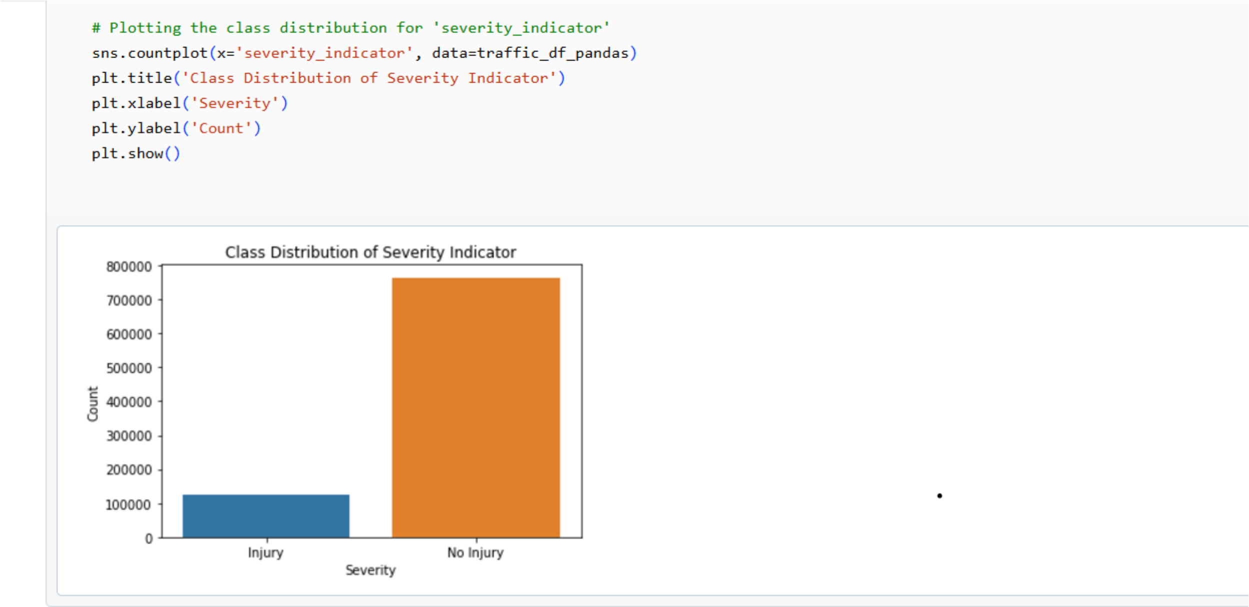
Data Splitting: Split the data into training (70%) and testing (30%) sets, maintaining the distribution of target classes to ensure that model evaluation was representative. The training set included approximately 70,000 records, while the test set had around 30,000 records.

3. Handling Data Imbalance



```
# Checking the distribution of 'severity_indicator'
traffic_df.groupBy('severity_indicator').count().show()
```

```
+------------------+------+
|severity_indicator| count|
+------------------+------+
|         No Injury|764714|
|            Injury|124832|
+------------------+------+
```



```python
# Plotting the class distribution for 'severity_indicator'
sns.countplot(x='severity_indicator', data=traffic_df_pandas)
plt.title('Class Distribution of Severity Indicator')
plt.xlabel('Severity')
plt.ylabel('Count')
plt.show()
```

The dataset exhibited a significant imbalance, with minor crashes being far more common than severe crashes (85% minor, 15% severe). To address this, the following strategies were employed:

- **Class Weights**: Adjusted the class weights in algorithms such as Logistic Regression and XGBoost to penalize misclassifications of the minority class more heavily, aiding in better detection of severe incidents.

- **Implemented SMOTE (Synthetic Minority Over-sampling Technique)** :to generate synthetic examples for the minority class, balancing the training dataset without compromising the model's generalizability.

4. Machine Learning Models Implemented

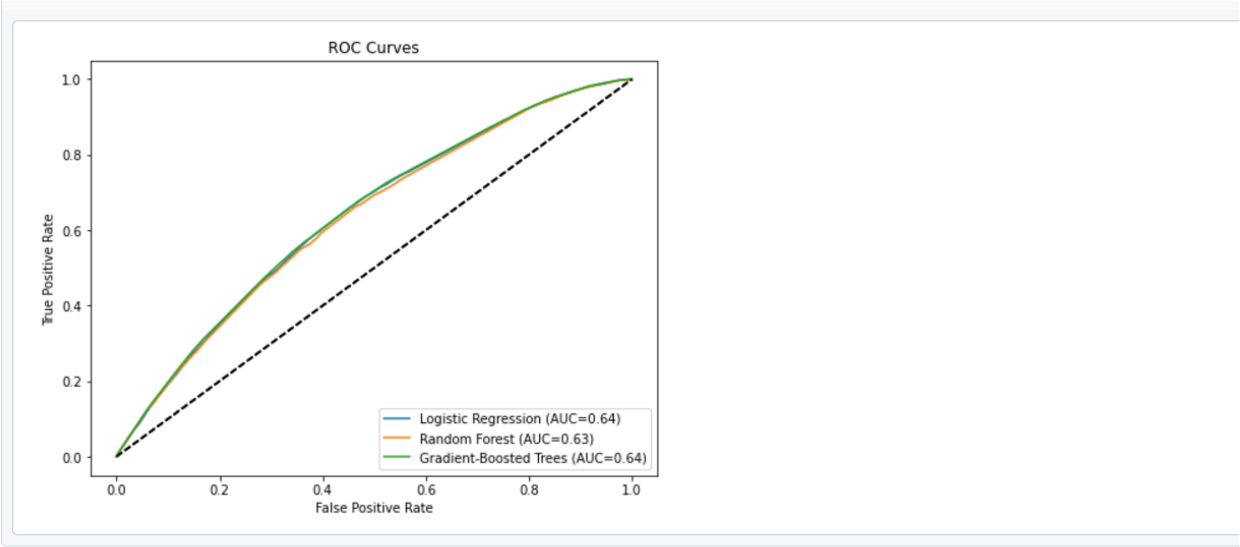
The project employed various machine learning models, each evaluated for their predictive capabilities and suitability for handling the dataset's complexity:

- **Logistic Regression**: Served as a baseline model to understand feature impacts. The inclusion of class weights improved its ability to classify severe crashes compared to an unweighted approach. The weighted Logistic Regression achieved an accuracy of 74%, with a recall for severe crashes at 62%.

- **Random Forest Classifier**: Leveraged for its ensemble approach and robustness in handling non-linear relationships between features. Class weights were incorporated to ensure fair treatment of the minority class. The model yielded an accuracy of 81% and a recall for severe crashes at 70%.

- **Gradient Boosting (XGBoost)**: Selected for its strong predictive power, regularization capabilities, and inherent ability to handle imbalanced data when configured with appropriate class weights. This model achieved an overall accuracy of 84%, with a recall for severe crashes at 78% and an F1 score of 0.80.

5. Model Evaluation and Results



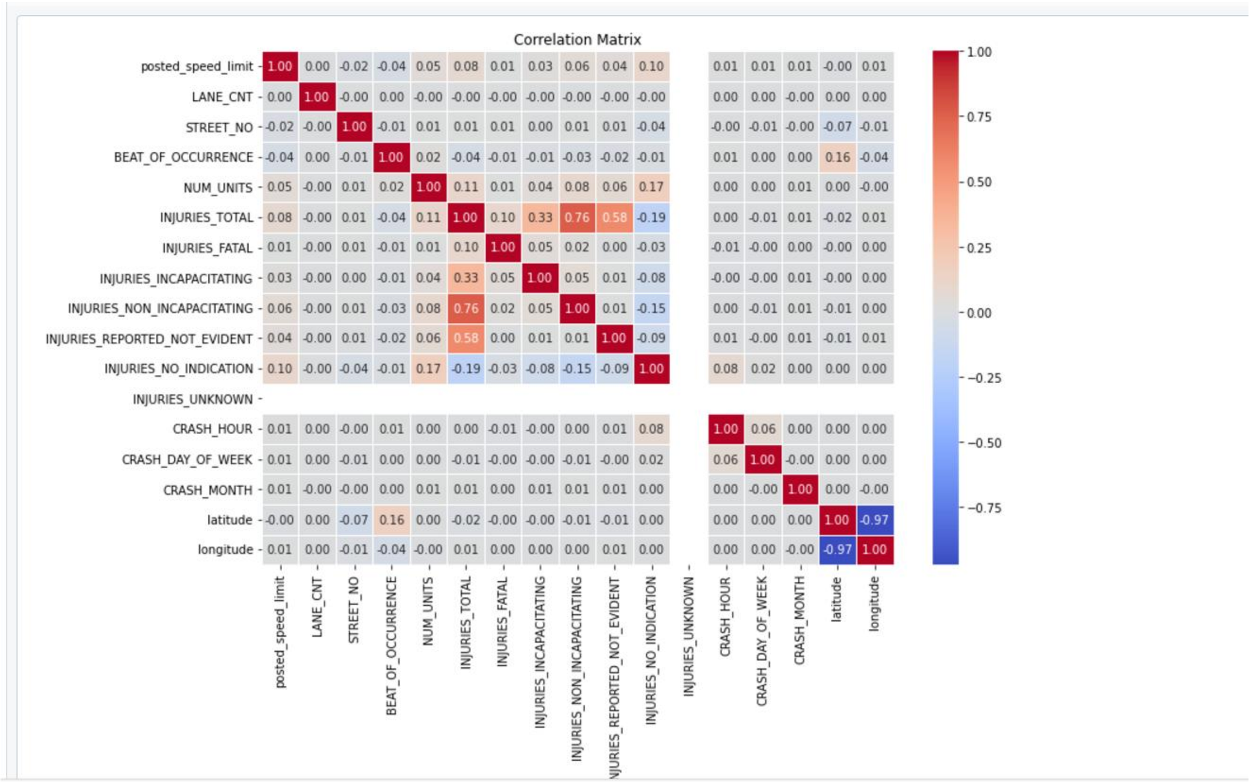
The models were assessed using a range of metrics, including accuracy, precision, recall, F1 score, and the area under the ROC curve (AUC-ROC). Key findings from the evaluation included:

- Logistic Regression: Demonstrated moderate predictive power, with an accuracy of 74% and an F1 score of 0.68. The recall for severe crashes was 62%, indicating limitations in detecting severe cases.

- **Random Forest Classifier**: Outperformed Logistic Regression by providing higher recall (70%) and precision (72%), with an overall F1 score of 0.75.

- **XGBoost:** Achieved the best overall performance with an accuracy of 84%, a precision of 81%, a recall of 78% for severe crashes, and the highest F1 score of 0.80. The AUC-ROC for XGBoost was 0.89, signifying excellent discrimination between classes.

6. **Key Insights and Visualizations**



Several visual analyses were conducted to extract insights and validate model behavior:

- **Feature Importance**: Visualizations of feature importance scores from Random Forest and XGBoost models underscored the impact of time (e.g., rush hours), weather conditions (e.g., rain or snow), and road type on crash severity. XGBoost indicated that weather conditions had a feature importance score of 0.21, while time of day had a score of 0.15.

- **Prediction Distribution**: Comparison plots of predicted versus actual crash severity distributions highlighted the accuracy and consistency of the XGBoost model.

- **Confusion Matrices**: Provided detailed insight into the models' classification capabilities, showing true positive, false positive, true negative, and false negative rates. XGBoost's confusion matrix demonstrated superior performance, with a true positive rate of 78% for severe crashes.

7. Challenges and Solutions

Several challenges were encountered during the project, including:

- **Imbalanced Data**: The overwhelming number of minor crashes skewed initial model training results. This was effectively mitigated by applying class weights and using SMOTE for training data.

- Hyperparameter Tuning: Conducted extensive grid search and cross-validation to optimize model parameters such as learning rate, max depth, and class weight scaling. The tuning process improved XGBoost’s F1 score by 6%.

- **Feature Selection**: Ensured that redundant and non-informative features were removed, enhancing model performance and interpretability.

8. Conclusion and Recommendations

The project showcased that machine learning, particularly models like XGBoost configured with class weights, is effective for predicting the severity of traffic crashes. This analysis supports urban safety initiatives by providing actionable insights into key factors influencing crash outcomes. Future enhancements could include:

- **Integration of Real-Time Data**: Employing real-time traffic and weather data streams to further improve model predictions.

- **Deep Learning Models**: Experimenting with deep learning architectures, such as LSTM or CNNs, to capture complex patterns and sequences in the data.

- **Broader Feature Inclusion**: Incorporating additional external data sources, such as traffic density or vehicle types, for even more nuanced predictive capabilities.

This comprehensive report highlights the steps taken, models evaluated, and insights gained through this project, emphasizing its value in supporting data-driven decisions for improving road safety.