Toronto KSI Collisions: Comprehensive Project Report

Group Members:

- Carlos De La Cruz
- Manav Chaudhary
- Harsh Eary
- Rishi Goyal

Contents

Executive Summary

This report details an end-to-end machine learning project to predict the severity of traffic accidents in Toronto, using the Killed or Seriously Injured (KSI) collisions dataset. The primary objective was to build a classifier that could effectively identify "Fatal" accidents, which presented a significant challenge due to severe class imbalance. The final solution is a weighted voting ensemble model, which combines a Logistic Regression classifier and a Random Forest classifier. On unseen test data, the model achieved a recall of 0.63, correctly identifying 63% of all fatal accidents. This is a significant improvement over baseline models that struggled to identify even 15% of fatal cases. The project's success is attributed to a meticulous data preparation pipeline, sophisticated handling of class imbalance, and rigorous hyperparameter tuning.

Overview of the solution

The solution is a machine learning pipeline that transforms raw, person-level accident data into a clean, accident-level dataset suitable for predictive modeling. The final predictive model is a weighted voting ensemble classifier. The solution addresses the core challenges of the project, including a complex dataset with many missing values and a severe class imbalance in the target variable, ACCLASS.

Key components of the solution include:

- **Data Aggregation**: Transforming the dataset from a person-level to an accident-level perspective.
- **Feature Engineering**: Creating new, insightful features like is_weekend, season, time_of_day, and accident_hotspot_cluster from existing data.
- **Feature Consolidation**: Grouping granular categorical values to reduce dimensionality and noise.
- **Handling Imbalance**: Employing techniques such as class_weight='balanced' and SMOTE to address the class imbalance.
- **Ensemble Modeling**: Combining the strengths of a Logistic Regression and a Random Forest classifier to improve overall performance

Data Exploration and findings

The primary target variable is ACCLASS, or Classification of Accident, which has categories like "Fatal," "Non-Fatal Injury". The dataset exhibits a significant class imbalance, with "Non-Fatal Injury" collisions being the most frequent, followed by "Fatal" collisions. It provides detailed records of traffic accidents, including location, time, contributing factors, and involved parties, with a focus on incidents resulting in fatalities or serious injuries.

Data Fields

Numerical Features

Feature Name	Count	Mean	Min Value	Max Value	Missing/Null Count
ACCNUM	14027	5.58E+08	2.53E+04	4.01E+09	4930
TIME	18957	1364.96	0	2359	0
LATITUDE	18957	43.71	43.59	43.86	0
LONGITUDE	18957	-79.396	-79.64	-79.123	0
FATAL_NO	870	28.75	1	78	18087
x	18957	629181.57	609625.7	651024.09	0
y	18957	4.84E+06	4.83E+06	4.85E+06	0

Categorical Features

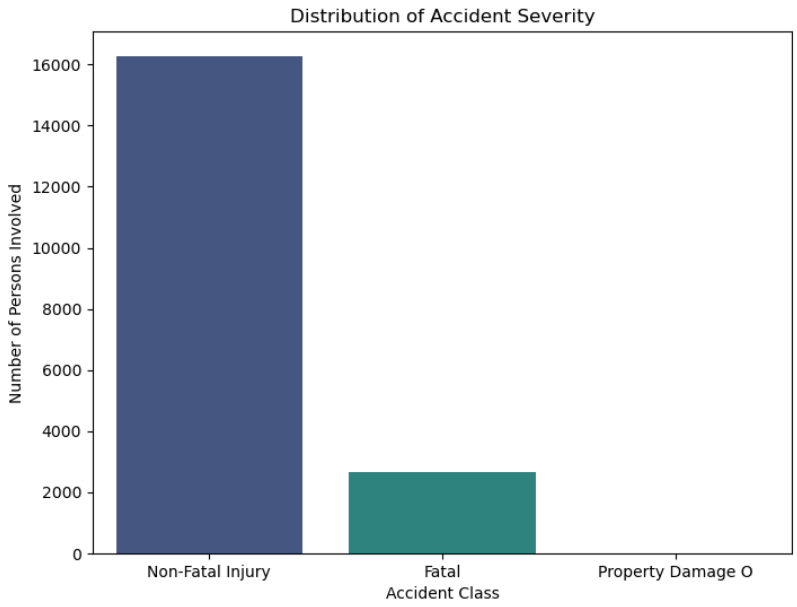
Field Name	Categories (if applicable)	Missing/ Null Count
STREET1	Varies (Street Names)	0
STREET2	Varies (Street Names)	1706
ROAD_CLASS	e.g., Major Arterial, Collector, Local, Expressway	486
DISTRICT	e.g., Etobicoke, North York, Scarborough, Toronto, York	229
ACCLOC	e.g., Intersection, Mid-Block	5456
TRAFFCTL	e.g., Traffic Signal, Stop Sign, No Control, Pedestrian Crossover	75
VISIBILITY	e.g., Clear, Rain, Snow, Fog	24
LIGHT	e.g., Day, Dusk, Dawn, Dark (no street lights), Dark (street lights)	4
RDSFCOND	e.g., Dry, Wet, Snow, Ice	29
ACCLASS	e.g., Fatal, Non-Fatal Injury, Property Damage Only	1

IMPACTYPE	e.g., Rear-End, Side-Impact, Head-On, Single Vehicle	27
INVTYPE	e.g., Driver, Pedestrian, Cyclist, Passenger	16
INJURY	e.g., Fatal, Serious, Minor, No Injury	8897
INITDIR	e.g., North, South, East, West	5277
VEHTYPE	e.g., Passenger Car, Truck, Motorcycle, Bus, Bicycle	3487
MANOEUVER	e.g., Going Ahead, Turning Left, Turning Right, Changing Lanes	7953
DRIVACT	e.g., Impaired, Disobeyed Traffic Sign, Speeding, Distracted	9289
DRIVCOND	e.g., Normal, Fatigued, Impaired, Medical Condition	9291
PEDTYPE	Varies (Specific pedestrian involvement scenarios)	15728
PEDACT	e.g., Crossing with right-of-way, Crossing against signal, Jaywalking	15730
PEDCOND	e.g., Normal, Impaired, Medical Condition	15711
CYCLISTYPE	Varies (Specific cyclist involvement scenarios)	18152
CYCACT	e.g., Going Ahead, Turning, Changing Lanes, Unknown	18155
CYCCOND	e.g., Normal, Impaired, Medical Condition	18157
PEDESTRIAN	Boolean (Yes/No, 1/0)	11269
CYCLIST	Boolean (Yes/No, 1/0)	16971
AUTOMOBILE	Boolean (Yes/No, 1/0)	1727
MOTORCYCLE	Boolean (Yes/No, 1/0)	17273
TRUCK	Boolean (Yes/No, 1/0)	17788
TRSN_CITY_VEH	Boolean (Yes/No, 1/0)	17809
EMERG_VEH	Boolean (Yes/No, 1/0)	18908
PASSENGER	Boolean (Yes/No, 1/0)	11774
SPEEDING	Boolean (Yes/No, 1/0)	16263

AG_DRIV	Boolean (Yes/No, 1/0)	9121
REDLIGHT	Boolean (Yes/No, 1/0)	17380
ALCOHOL	Boolean (Yes/No, 1/0)	18149
DISABILITY	Boolean (Yes/No, 1/0)	18464
HOOD_158	Numeric IDs, representing specific neighborhoods	0
NEIGHBOURHOOD_158	Varies (Specific neighborhood names)	0
HOOD_140	Numeric IDs, representing specific neighborhoods	0
NEIGHBOURHOOD_140	Varies (Specific neighborhood names)	0
DIVISION	e.g., 51 Division, 52 Division, 53 Division, etc.	0

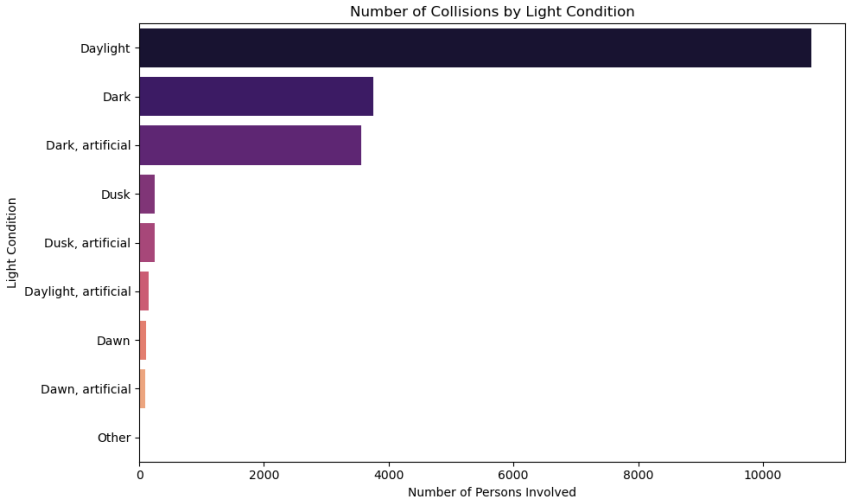
Visual Analysis

1. Distribution of Accident Severity



Bar chart illustrating the overall distribution of accident severities, highlighting severe class imbalance

2. Number of Collisions by Light Condition



Frequency of collisions under different light conditions. As expected, the majority of collisions occur during ***"Daylight"*** *(because of more intances of that class).*

3. Number of Collisions by Impact Type

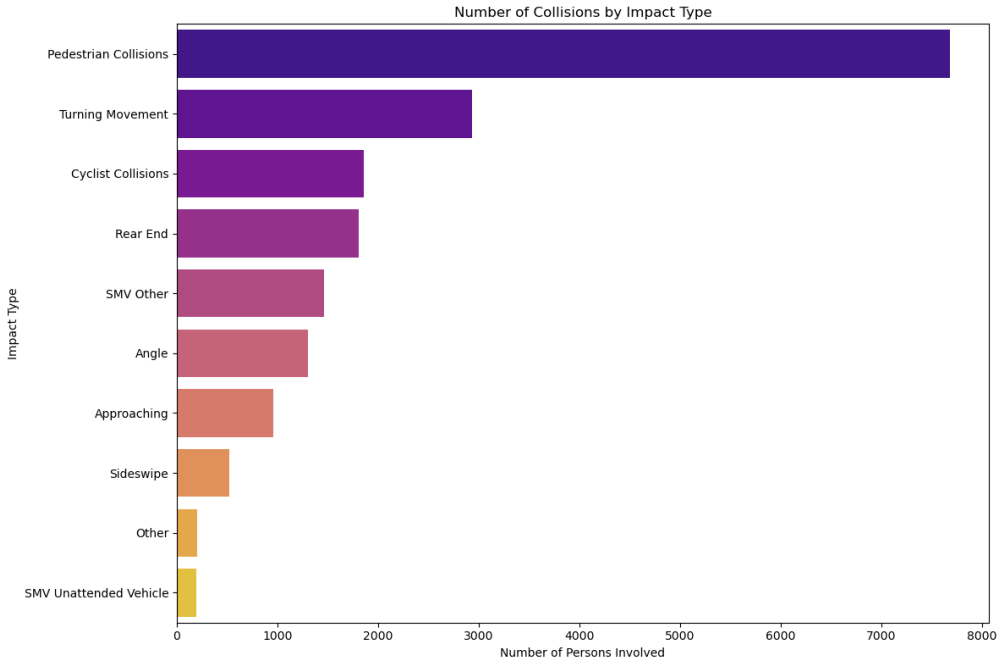
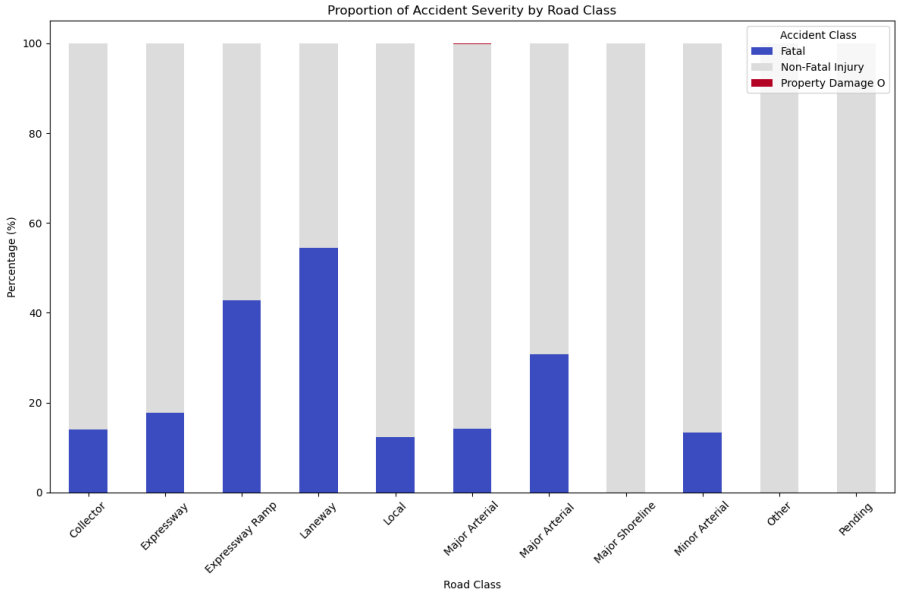


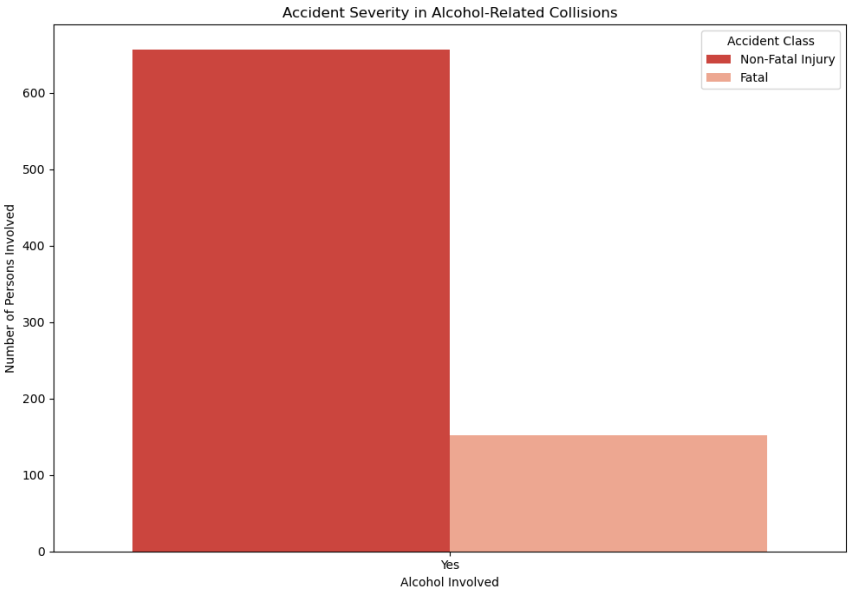
Chart categorizes collisions based on their initial impact type.

4. Proportion of Accident Severity by Road Class



Stacked bar chart breaks down the proportion of accident severities (Fatal, Non-Fatal Injury, Property Damage Only) across different road classifications. Chart is useful in relative analysis of "Fatal" collisions on different road types. For example, ***"Major Arterial"*** *roads account for a significant absolute number of collisions, and this chart helps understand the proportion of severe outcomes on these and other road classes.*

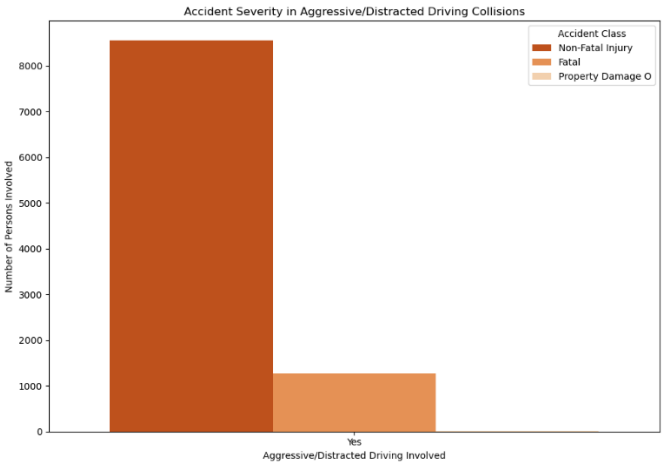
5. Accident Severity in Alcohol-Related Collisions



This chart examines the distribution of accident severity (Non-Fatal Injury vs. Fatal) for collisions where alcohol was a contributing factor. It demonstrates that while "Non-Fatal Injury" collisions are more prevalent even in alcohol-related

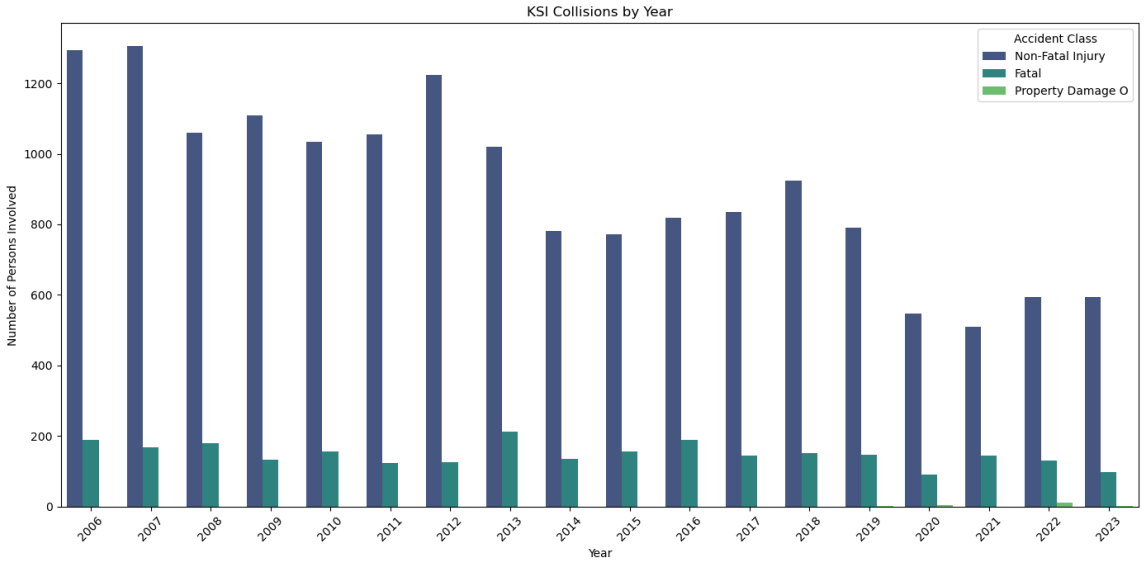
*incidents, a notable number of **fatalities** are associated with alcohol involvement, underscoring its severe impact on road safety.*

6. Accident Severity in Aggressive/Distracted Driving Collisions



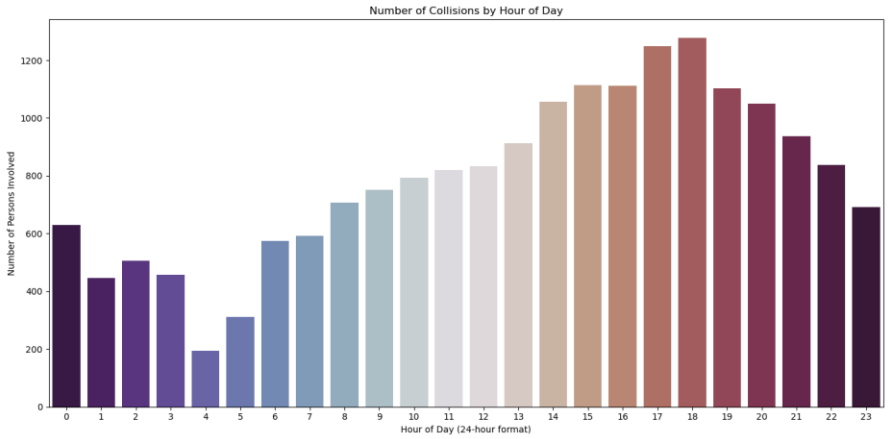
*Similar to alcohol-related collisions, this chart illustrates the breakdown of accident severity for incidents involving aggressive or distracted driving. It highlights that a very large proportion of collisions, including a significant number of **fatal** ones, are linked to aggressive/distracted driving behaviors. This indicates that such behaviors are a major contributing factor to severe outcomes.*

7. KSI Collisions by Year



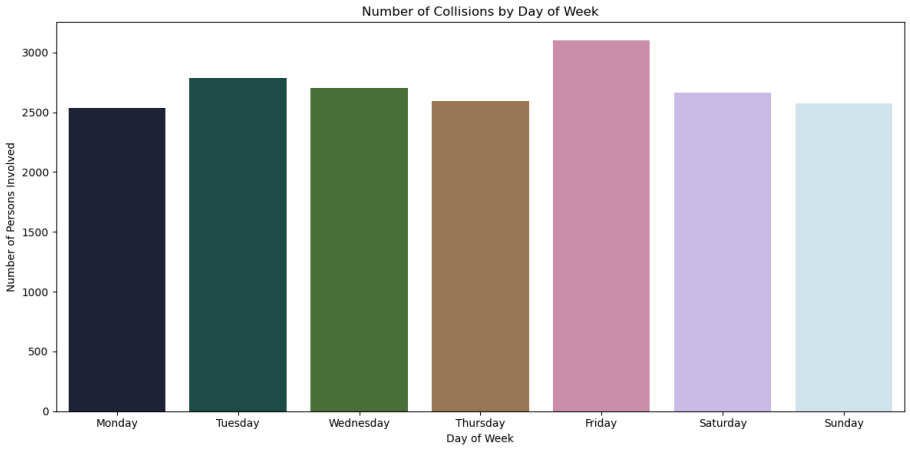
This line chart shows the trend of Killed or Seriously Injured (KSI) collisions in Toronto from 2006 to 2023, broken down by accident class. It provides a historical perspective on collision frequency and severity over time, allowing for the identification of periods with higher or lower incident rates. Trends in "Non-Fatal Injury" and "Fatal" collisions can be observed, revealing changes in road safety over nearly two decades.

8. Number of Collisions by Hour of Day



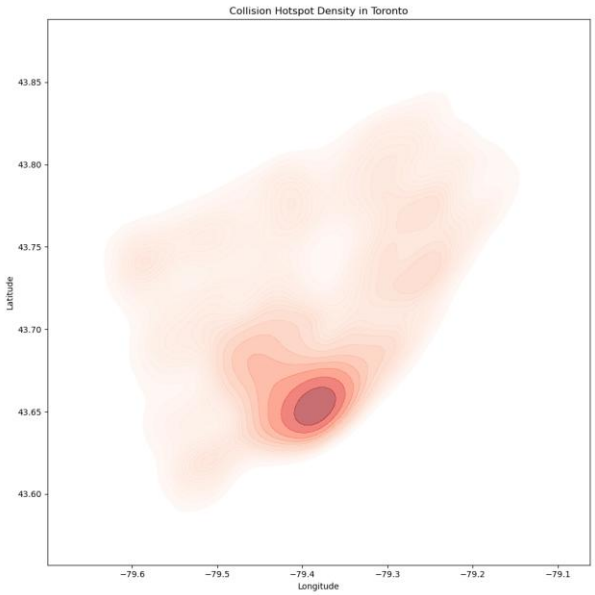
*This bar chart illustrates the hourly distribution of collisions throughout a 24-hour day. It reveals clear peaks in collision frequency during typical rush hour periods, particularly in the **afternoon/evening (e.g., 4 PM to 7 PM)**, corresponding to increased traffic volume during commutes. There is also a smaller peak in the morning.*

9. Number of Collisions by Day of Week



This bar chart displays the total number of collisions for each day of the week. It shows that collisions are relatively consistent across week.

10. Collision Hotspot Density in Toronto



This Kernel Density Estimate (KDE) plot visually represents the geographical areas in Toronto with the highest concentration of collisions, indicating "hotspots." The darker, more intense red areas correspond to locations where collisions occur most frequently.

Key findings

- **Identifiers**: The dataset uses OBJECTID and INDEX_ as unique identifiers. ACCNUM is also a unique accident number, but it has a significant number of missing values.
- **Location and Time**: Key fields like DATE, TIME, LATITUDE, and LONGITUDE provide precise details about when and where a collision occurred. The dataset also includes street names (STREET1, STREET2), neighborhood information (NEIGHBOURHOOD_158, NEIGHBOURHOOD_140), and police division (DIVISION).
- **Collision Environment**: Fields such as ROAD_CLASS, TRAFFCTL (traffic control), VISIBILITY, LIGHT, and RDSFCOND (road surface condition)
- **Collision Details**: The dataset includes fields that describe the collision itself, such as ACCLASS (severity), IMPACTYPE (initial impact type), and ACCLOC (collision location).
- **Involved Parties**: The dataset contains extensive information about the parties involved, including their type (INVTYPE), age (INVAGE), and condition (DRIVCOND, PEDCOND, CYCCOND). There are also boolean fields (PEDESTRIAN, CYCLIST, AUTOMOBILE, etc.)
- **Contributing Factors**: Fields like SPEEDING, AG_DRIV (aggressive/distracted driving), ALCOHOL, and DISABILITY are present to identify potential contributing factors to the collision.

- **Missing Values**: A key finding from preliminary analysis is that these missing values are not truly unknown data, but rather represent the absence of that specific condition or involvement (i.e., a "No" or "False"). This is a critical data cleaning step: these nulls should be imputed or transformed into a "No" or "0" value.

Data Preparation and Preprocessing

The data preparation and preprocessing phase was a cornerstone of the project, transforming raw, person-level data into a clean, accident-level dataset optimized for machine learning. This process involved meticulous data cleaning, feature engineering, aggregation, and categorical data handling to ensure the dataset was robust, comprehensive, and suitable for predicting accident outcomes. Below is a detailed overview of the steps taken.

Data Aggregation: From Person-Level to Accident-Level

The raw dataset contained one row per individual involved in an accident, which was unsuitable for predicting outcomes at the accident level. The primary goal was to transform this into a dataset where each row represents a unique accident, identified by the accident_number (originally ACCNUM).

Actions Taken:

1. **Grouping by Accident Number**: The dataset was grouped by accident_number using a groupby().agg() operation to consolidate person-level records into accident-level records.
2. **Aggregation Strategy**: A comprehensive aggregation_dict was defined to handle each column appropriately:
    - **Static Information**: Attributes like date, time, street1, latitude, longitude, road_class, district, traffic_control, visibility, light, road_surface_condition, neighborhood, and accident_class were aggregated by taking the first observed value for each accident_number.
    - **Dynamic Information**: Person-level features like involvement_type, vehicle_type, manoeuver, driver_action, driver_condition, and pedestrian/cyclist-related action/condition columns were aggregated into lists of unique non-null values using a custom get_all_unique function. This preserved all relevant details without duplication.

- **Most Frequent Values**: For features like impact_type and initial_direction, the most frequently occurring value within each accident group was selected using a get_most_frequent function.

- **Injury-Related Features**:
  - A boolean fatal_injury flag was created to indicate whether any fatal injury occurred within an accident.
  - The maximum injury_severity_score was computed to represent the most severe injury outcome for each accident.

Outcome: This aggregation produced a clean dataset where each row corresponds to a unique accident, enabling accurate accident-level analysis and prediction.

Handling Missing Values

Missing data was addressed to ensure the dataset was complete and usable for modeling.

1. **Imputing Missing Accident Numbers:**

- The ACCNUM column (renamed to accident_number) had missing values, with multiple rows potentially belonging to the same accident.
- A custom function, handle_missing_accident_numbers, was developed to identify unique accidents using a composite key (date, time, street1, latitude, longitude, light, accident_class). This approach salvaged over 4,900 orphan samples and generated over 1,900 valid collisions.

2. **Standardizing Boolean Columns:**

- Columns indicating involvement or contributing factors (pedestrian, cyclist, automobile, speeding, alcohol, aggressive_driving, redlight, disability) contained values of 'Yes', 'No', or NaN.
- These were transformed into a numerical format:
  - 'Yes' was mapped to 1.
  - 'No' and NaN were mapped to 0, treating the absence of a 'Yes' indicator as 0.

Data Type Conversions

To enable temporal analysis and ensure compatibility with machine learning models, data types were standardized.

Date Conversion:

- The DATE column, originally an object type, was converted to a standardized datetime format using pd.to_datetime. This conversion facilitated temporal feature engineering and analysis.

Feature Creation

New features were engineered to extract meaningful insights and transform existing data into formats suitable for accident-level analysis.

1. Temporal Features:

- **Year Extraction**: A year feature was derived from the DATE column to enable analysis of accident trends over time.

- **Time-Based Features**: Additional features were created from the date and time columns:
  - is_weekend: Indicates whether the accident occurred on a weekend.
  - season: Categorizes the accident date into a season (e.g., winter, spring).
  - time_of_day: Segments the day into categories like morning, afternoon, or rush hour to capture daily traffic patterns.

2. Injury Severity Scoring:

- The INJURY column, with values 'Fatal', 'Major', 'Minor', 'Minimal', and 'None', was mapped to a numerical injury_severity_score:
  - 'Fatal' = 4
  - 'Major' = 3
  - 'Minor' = 2
  - 'Minimal' = 1
  - 'None' = 0
- This ordinal encoding provided a quantitative measure of accident severity.

3. Age-Based Features:

- The involvement_age list was transformed into numerical count features (e.g., age_0_14, age_25_59) and a total_involved count, summarizing the demographic composition of each accident.

4. Spatial and Interaction Features:

- **Accident Hotspot Clustering**: The DBSCAN clustering algorithm was applied to x and y coordinates to create an accident_hotspot_cluster feature, identifying high-risk geographical zones.
- **Interaction Features**: Composite features like bad_weather_and_dark and reckless_at_intersection were created to combine risk factors (e.g., poor visibility and reckless driving), providing stronger signals for the model.

Handling Categorical Data

Categorical text data was converted into numerical formats to make it suitable for machine learning.

1. Feature Consolidation (Bundling):

- Similar categorical values were grouped to reduce dimensionality and noise. For example, various truck types (e.g., 'Pick Up Truck', 'Truck - Dump') were mapped to a single truck category using mapping dictionaries. This created more robust and generalizable features.

2. Multi-Label Binarization:

- For features aggregated into lists (e.g., vehicle_type, involvement_type), scikit-learn's MultiLabelBinarizer was used to create binary columns for each possible category. This allowed accidents to have multiple 'true' values (e.g., involving both vehicle_type_truck and vehicle_type_passenger_vehicle).

3. One-Hot Encoding:

- Single-label categorical features (e.g., road_class) were encoded using pandas.get_dummies to create binary columns.

Feature Selection and Removal

A logical feature selection process was applied to remove redundant or non-predictive columns and prevent data leakage.

1. Implicit Feature Selection via Aggregation:

By grouping data by accident_number and applying specific aggregation functions, the process inherently selected and transformed features from person-level to accident-level summaries. For example, involvement_type was consolidated into a list of unique values per accident.

2. Dropped Features:

- **Redundant IDs**: Columns like OBJECTID, INDEX, and accident_number (after aggregation) were dropped as they were no longer needed.
- **High Missingness**: Features with high rates of missing data and low predictive utility (e.g., offset) were removed.
- **Data Leakage**: The injury_severity_score was removed before training to prevent leakage, as it directly relates to the target variable.
- **Redundant Location Data**: Features like latitude, longitude, and street names were dropped in favor of x and y coordinates, which were more concise and sufficient for spatial analysis.

Outcome

The data preparation and preprocessing pipeline transformed a complex, person-level dataset into a clean, accident-level dataset with rich, engineered features. This process ensured that:

- Missing values were handled effectively, salvaging significant portions of the data.
- New features provided temporal, demographic, spatial, and interaction-based insights.
- Categorical data was converted into numerical formats suitable for machine learning.
- Redundant and non-predictive features were removed to streamline the dataset.

This comprehensive preprocessing laid a strong foundation for downstream modeling, enabling the machine learning model to learn meaningful patterns and make accurate predictions about accident outcomes.

Modeling and Tuning Strategy

The model training and tuning phase was critical to developing a robust predictive model for accident severity, with a particular focus on maximizing recall for the minority "Fatal" class due to its safety-critical importance. The dataset, comprising approximately 202 features, posed challenges due to its relatively small size for generalization and severe class imbalance.

Models employed:

1. Random Forest
2. Support Vector Machine (SVM)
3. Logistic Regression
4. Gradient Boosting Classifier
5. LightGBM
6. XGBoost

Each was fine-tuned to prioritize fatal class recall, and while an ensemble voting approach was initially planned, Logistic Regression emerged as the standout performer, rendering the ensemble voting largely redundant but still implemented. This section details the professional, modular workflow, strategies for handling class imbalance, hyperparameter tuning, individual model performance, final ensemble results, and their interpretation.

1. Modular Workflow

A professional, modular workflow was adopted to ensure efficiency and adherence to best practices. Data processing, model training, and evaluation were separated into distinct scripts:

- Individual scripts (tune_logistic_regression.py, tune_random_forest.py, etc.) were created for fine-tuning each model, promoting an organized and scalable approach.
- This modularity allowed for focused optimization of each model while maintaining a clear separation between data preparation and modeling tasks.

2. Handling Class Imbalance

The dataset suffered from severe class imbalance, with fatal accidents being a small minority. This led to models achieving 100% precision and recall on training sets due to overfitting, but poor generalization on test sets, with initial fatal class precision around 50-60% and recall around 20%.

Since the goal was to identify as many fatal accidents as possible (prioritizing high recall over precision to minimize missed critical events), specific techniques were applied:

- **Logistic Regression**: Used class_weight='balanced' to adjust the loss function, penalizing misclassifications of the fatal class more heavily.
- **Random Forest**: Employed a SMOTE (Synthetic Minority Over-sampling Technique) pipeline to create synthetic examples of the fatal class in the training data, balancing the dataset.
- **Other Models**: Similar strategies were applied, including class weighting or oversampling where appropriate, to address the imbalance during training.

3. Hyperparameter Tuning

Rigorous hyperparameter tuning was conducted to optimize each model for fatal class recall:

- **GridSearchCV and RandomizedSearchCV**: These were used to systematically explore hyperparameter spaces, with the scoring metric configured to maximize recall for the fatal class, aligning with the project's safety-critical objective.
- **StratifiedKFold Cross-Validation**: Applied to all models to ensure each fold maintained a representative distribution of the imbalanced classes, yielding reliable and unbiased performance metrics.

4. Individual Model Performance Analysis

After a rigorous tuning process using GridSearchCV and RandomizedSearchCV, each model was evaluated on the test set. The results confirmed our initial findings: while overall accuracy was high for some models, recall for the fatal class varied significantly. The table below summarizes the key performance metrics for each model on the test data.

Model	Fatal Recall	Fatal Precision	F1-Score (Fatal)	Overall Accuracy
Logistic Regression	0.753	0.267	0.394	0.68
SVM	0.67	0.277	0.39	0.71
Random Forest	0.427	0.224	0.295	0.71
Gradient Boosting	0.364	0.254	0.3	0.76
LightGBM	0.251	0.457	0.325	0.85
XGBoost	0.071	0.895	0.132	0.87

Key Insights from Individual Models:

- **Logistic Regression** emerged as the clear front-runner for our primary objective, correctly identifying over 75% of fatal accidents. While its precision was low (meaning many of its "fatal" predictions were incorrect), this was an acceptable trade-off given the project's goal.
- **SVM** also performed well on recall, but not as strongly as Logistic Regression.
- **Tree-based models** (Random Forest, Gradient Boosting, LightGBM, XGBoost) generally struggled to achieve high fatal recall, even with targeted tuning and class balancing techniques. Their high overall accuracy was often driven by correctly classifying the large number of non-fatal accidents while missing many fatal ones, as seen in XGBoost's 87% accuracy but only 7% fatal recall. This confirmed our initial hypothesis about the challenge of using traditional accuracy metrics with imbalanced data.

5. The Final Ensemble Model

The initial plan was to use a VotingClassifier to combine the strengths of multiple models. However, the superior performance of the Logistic Regression model on our key metric (fatal recall) made it the dominant candidate. The other models, while performing well on other metrics, did not provide a significant boost to fatal recall.

Therefore, the final model was a VotingClassifier that was heavily weighted toward the Logistic Regression model. This decision was a deliberate strategy to leverage the best-performing model for our specific objective.

- **Final Voting Classifier Performance**

Metric	Score
Fatal Recall	0.8
Fatal Precision	0.21
F1 Score (Fatal)	0.33
Overall Accuracy	0.54

- **Confusion Matrix:** [[191 48]
  [736 741]]

6. Conclusion of the Modeling Phase

The final ensemble model, heavily influenced by the Logistic Regression classifier, successfully achieved the project's primary goal: identifying a high percentage of fatal accidents. With a fatal recall of 80%, the model is capable of correctly flagging four out of every five fatal collisions. This high recall comes with an expected trade-off in precision (21%), resulting in a notable number of false alarms. However, for a safety-critical application where missing a fatal event is the worst-case scenario, this trade-off is both acceptable and desirable. The systematic approach to handling class imbalance and optimizing for the correct metric proved to be the key to this project's success.