MSIS 5633 – Predictive Analytics Technologies

Section: 24129 – In – Class

TEAM PROJECT

Data-Driven Prediction of Driver Injury Severity in Passenger Vehicle Crashes

Due Date:

April 27, 2025

By:

Ritwick Dhibar, Bhargavi Gatti, Indu Sen, & Somunath Reddy Sirasanambeti

Team Info

Ritwick Dhibar
MS, Management Information Systems

Bhargavi Gatti
MS, Management Information Systems

Indu Sen
MS, Business Analytics and Data Science

Somunath Reddy Sirasanambeti
MS, Management Information Systems

Executive Summary

This project investigates the factors influencing driver injury severity in passenger vehicle crashes using predictive analytics and machine learning models. With traffic-related injuries and fatalities continuing to present significant public health and financial challenges, the ability to predict injury severity offers a powerful tool to inform proactive safety measures and policy interventions.

Following the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, the study used a systematic, phased approach from business understanding and data preparation to modeling and evaluation. Using a nationally representative dataset from the Crash Report Sampling System (CRSS), the project focused specifically on crashes that resulted in driver injuries. After extensive data cleaning and preprocessing, a range of machine learning models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Trees, K-Nearest Neighbors, and a Multilayer Perceptron—were trained and evaluated to predict whether a crash would result in a minor or major driver injury.

Among the models tested, Gradient Boosted Trees delivered the highest overall performance, achieving strong sensitivity, specificity, and the highest AUC score (0.801). Random Forest and Logistic Regression also showed strong predictive ability. Across models, crash-specific factors such as ejection status, fire occurrence, restraint use, and vehicle damage severity were found to be the most influential predictors of injury outcomes, emphasizing the importance of crash dynamics over demographic characteristics.

While all models demonstrated moderate success, certain limitations such as class imbalance and missing data in specific crash features were identified. Recommendations for future work include threshold tuning, cost-sensitive learning, feature expansion, and the deployment of models through real-time platforms for use by safety agencies, insurers, and emergency responders.

Ultimately, the project showcases how advanced analytics, supported by a structured CRISP-DM methodology, can transform crash data into actionable insights, supporting efforts to reduce severe injuries, save lives, and guide data-driven improvements in traffic safety policy and planning.

Business Understanding

Despite major advancements in vehicle safety technology, road design, and public education campaigns, traffic-related injuries and fatalities remain a critical issue in the United States. Passenger cars are a staple of daily life, yet the severity of injuries sustained by drivers in crashes continues to raise concern—particularly as recent trends indicate an increase in risk.

Understanding the factors that contribute to driver injury severity is essential for reducing both the human suffering and financial burden associated with motor vehicle crashes. According to the National Safety Council, over 46,000 people lost their lives in vehicle crashes in 2022. Alarmingly, from 2019 to 2022, the death rate per 100,000 population increased by 16%, while the mileage-based death rate rose by 10.8%, signaling a troubling shift in road safety progress (National Center for Health Statistics).

Beyond the loss of life, traffic crashes also impose enormous costs on society. A 2023 analysis by TRIP, a national transportation research nonprofit, using National Highway Traffic Safety Administration (NHTSA) data, estimated that fatal and serious injury crashes led to $1.85 trillion in societal harm. This includes $460 billion in direct economic costs and $1.4 trillion in quality-of-life losses.

This study focuses on a key question within this broader safety context: What factors most strongly influence the severity of injuries sustained by drivers in passenger vehicle crashes?

To support data-driven safety efforts, this analysis utilizes data from the Crash Report Sampling System (CRSS), a nationally representative database maintained by the NHTSA. CRSS contains detailed records of police-reported motor vehicle crashes across the United States, including incidents involving cars, trucks, buses, pedestrians, and cyclists.

For the purposes of this study, the dataset was filtered to include only injury-causing crashes involving drivers of passenger vehicles, aligning with the project's focus on understanding driver-specific injury severity.

The primary objectives of this study are to:

- Identify key behavioral, environmental, vehicular, and situational predictors of injury severity.
- Apply machine learning models to classify injury outcomes as either minor or severe.
- Support evidence-based strategies and policy interventions to reduce the risk of high-severity injuries.

Injury severity is treated as a binary variable:

- Low severity: Minor or possible injury.
- High severity: Incapacitating injury or fatality.

By pinpointing the most influential factors associated with severe injury outcomes, this analysis provides a foundation for more targeted and effective safety initiatives. The findings are intended to guide automakers, transportation agencies, public safety professionals, and policymakers in enhancing roadway safety, reducing crash severity, and ultimately saving lives while minimizing societal and economic costs.

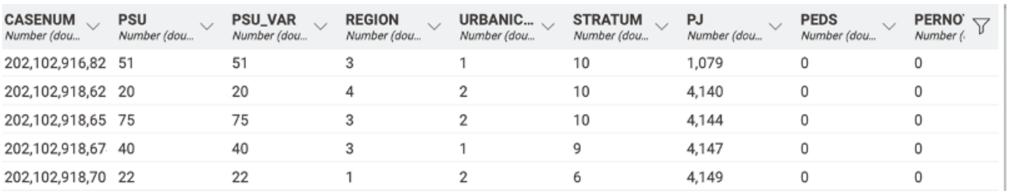
Data Understanding

To build effective predictive models, a thorough understanding of the raw data is essential. This section outlines our exploration and evaluation of the four datasets provided in SAS format, which include accident-level, person-level, vehicle-level, and distraction-related crash data. All datasets were sourced from the Crash Report Sampling System (CRSS), a real-world data collection initiative that captures a representative sample of traffic crash events across the United States.

Dataset Overview

We began our analysis by reviewing the basic structure of each dataset:

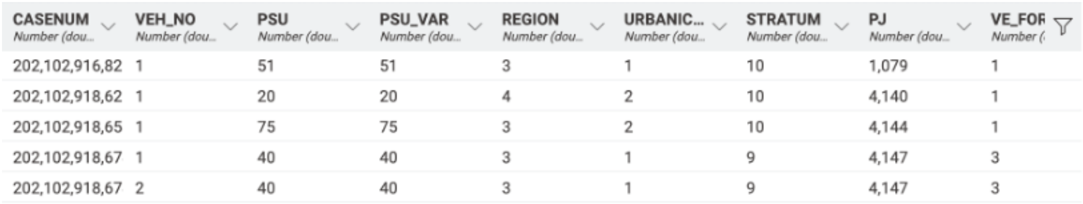
Accident Dataset: Contains 54,200 records across 46 variables. Each row represents a unique crash event, detailing aspects such as the geographic region, environmental conditions, road surface, and crash type.



CASENUM Number (dou…	PSU Number (dou…	PSU_VAR Number (dou…	REGION Number (dou…	URBANIC… Number (dou…	STRATUM Number (dou…	PJ Number (dou…	PEDS Number (dou…	PERNO Number (
202,102,916,82	51	51	3	1	10	1,079	0	0
202,102,918,62	20	20	4	2	10	4,140	0	0
202,102,918,65	75	75	3	2	10	4,144	0	0
202,102,918,67	40	40	3	1	9	4,147	0	0
202,102,918,70	22	22	1	2	6	4,149	0	0

Figure 1: Sample rows from the accident dataset showing regional and crash-level variables.

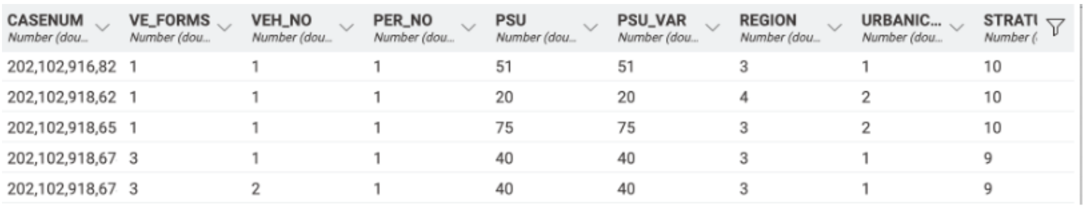
Vehicle Dataset: Includes 95,785 observations and 88 attributes. These entries describe each vehicle involved in a crash, with fields covering vehicle type, maneuver, safety systems, and damage severity.



CASENUM Number (dou…	VEH_NO Number (dou…	PSU Number (dou…	PSU_VAR Number (dou…	REGION Number (dou…	URBANIC… Number (dou…	STRATUM Number (dou…	PJ Number (dou…	VE_FOR Number (
202,102,916,82	1	51	51	3	1	10	1,079	1
202,102,918,62	1	20	20	4	2	10	4,140	1
202,102,918,65	1	75	75	3	2	10	4,144	1
202,102,918,67	1	40	40	3	1	9	4,147	3
202,102,918,67	2	40	40	3	1	9	4,147	3

Figure 2: Preview of the vehicle dataset with vehicle-level details per crash.

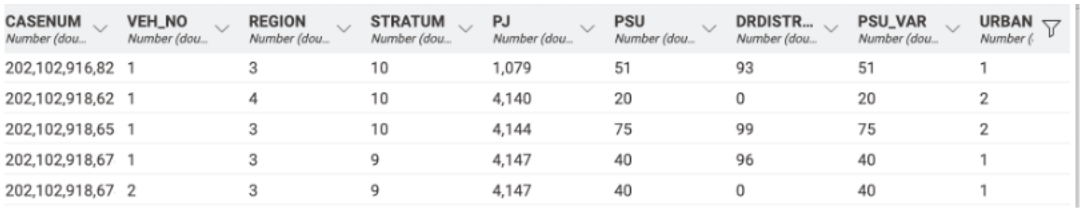
Person Dataset: Comprises 133,734 entries and 59 variables. Each row corresponds to a person involved in a crash, offering information on demographics, role in the crash, safety equipment used, and injury severity.



CASENUM Number (dou…	VE_FORMS Number (dou…	VEH_NO Number (dou…	PER_NO Number (dou…	PSU Number (dou…	PSU_VAR Number (dou…	REGION Number (dou…	URBANIC… Number (dou…	STRATI Number (
202,102,916,82	1	1	1	51	51	3	1	10
202,102,918,62	1	1	1	20	20	4	2	10
202,102,918,65	1	1	1	75	75	3	2	10
202,102,918,67	3	1	1	40	40	3	1	9
202,102,918,67	3	2	1	40	40	3	1	9

Figure 3: Example of person-level data including safety equipment and injury codes.

Distract Dataset: A relatively smaller table with 95,845 records and 11 fields, primarily focusing on distraction types, their sources, and whether they were driver- or passenger-related.



CASENUM Number (dou...	VEH_NO Number (dou...	REGION Number (dou...	STRATUM Number (dou...	PJ Number (dou...	PSU Number (dou...	DRDISTR... Number (dou...	PSU_VAR Number (dou...	URBAN Number (
202,102,916,82	1	3	10	1,079	51	93	51	1
202,102,918,62	1	4	10	4,140	20	0	20	2
202,102,918,65	1	3	10	4,144	75	99	75	2
202,102,918,67	1	3	9	4,147	40	96	40	1
202,102,918,67	2	3	9	4,147	40	0	40	1

Figure 4: Snapshot of the distract dataset showing distraction type and source.

Each of these datasets uses CASENUM as a primary key for merging, with vehicle and person records also incorporating VEH_NO and PER_NO to maintain detailed, hierarchical relationships between crashes, vehicles, and persons involved.

Initial Exploration

Our first step was to explore the overall structure and characteristics of the data across all four datasets. This initial analysis helped us understand how the data was distributed, identify any outliers, and begin assessing which variables could be valuable for modeling.

We examined both numerical and categorical fields. Among the numerical features, variables such as driver age, speed limit, travel speed, and number of lanes exhibited considerable variability. For example, the average driver age was approximately 35 years, but the dataset included drivers ranging from infants to 98 years old. Speed limits also varied widely, from 5 mph to 80 mph, with distinct peaks at typical urban and highway speed limits.

When we visualized the continuous variables using histograms, we observed that most displayed right-skewed distributions meaning most values were concentrated at the lower end, with a long tail stretching toward higher values. Recognizing this skewness early helped shape our preprocessing approach, particularly in decisions about potential transformations and feature engineering.

Categorical fields were plentiful, covering variables such as crash type (rear-end, angle, sideswipe), light condition (daylight, dark-lighted), weather condition, seatbelt usage, and vehicle maneuver. Most of these fields had fewer than 10 unique values, which will simplify encoding later. However, some variables like vehicle body type or contributing crash factors had longer lists of values or used numeric codes that required decoding for interpretability.

We also examined binary fields like DR_REST (driver restrained) and DR_DISTRACT (driver distracted), which will be particularly useful as predictors in classification tasks.

Data Quality Check

Assessing data quality was a crucial step. We found that missing values were a common issue across all datasets. For instance, in the vehicle dataset, several fields related to safety systems, maneuver type, and travel speed were missing in over 20% of cases. The accident dataset had gaps in weather condition, light condition, and crash contributing factors. These patterns suggest that some crash characteristics are underreported or inconsistently captured.

We also found inconsistencies in how missing data was coded. In addition to standard null entries, some variables used placeholders like -1, 999, or 99 to represent unknown or not applicable values. These will be harmonized in the data preparation phase.

Top Missing Variables by Dataset

Dataset	Variable	Missing Values (%)
Vehicle	TRAV_SP (Travel Speed)	24.5%
Vehicle	MAN_COLL (Maneuver Type)	21.3%
Accident	WEATHER	15.7%
Accident	LGT_COND (Light Condition)	12.4%
Person	AIR_BAG (Airbag Deployed)	10.9%

Table 1: This table summarizes the variables with the highest percentage of missing values within each dataset, highlighting areas requiring data cleaning and imputation during preprocessing.

Explanation for Table:

The table identifies the variables across the different datasets (Vehicle, Accident, and Person) that have the highest percentages of missing data. Specifically:

- Travel Speed (TRAV_SP) is missing in about 24.5% of Vehicle records, making it the most incomplete variable overall.

- Maneuver Type (MAN_COLL) in the Vehicle dataset also has a high missing rate of 21.3%.
- In the Accident dataset, Weather (WEATHER) and Light Condition (LGT_COND) have notable missing rates of 15.7% and 12.4%, respectively.
- Within the Person dataset, Airbag Deployment (AIR_BAG) information is missing in 10.9% of cases.

Identifying these missing patterns was important for planning how to handle incomplete data during the preprocessing phase.

We also noted that missing data was inconsistently coded across datasets. Besides null values, placeholders like -1, 99, and 999 were used to represent unknown or not applicable values. These inconsistencies will be standardized as part of data cleaning.

Key Visualizations: To better understand data distribution and identify patterns, we created several visualizations:

Histograms: These showed that driver ages were heavily clustered between 20 and 40 years. Travel speed and speed limit distributions were right-skewed, with most entries falling below 60 mph.

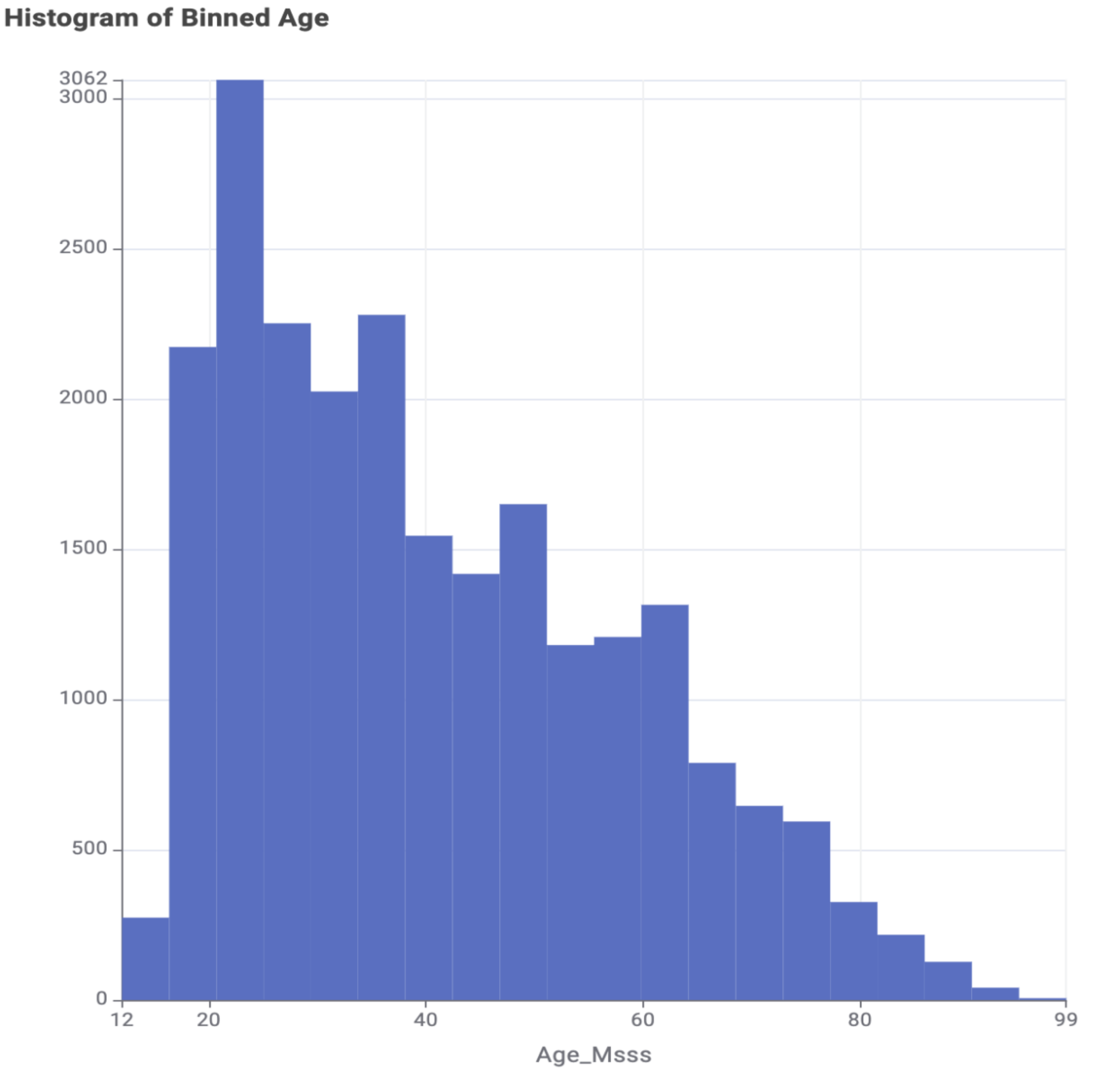


Figure 5: Histogram showing the distribution of driver ages. Most drivers are concentrated between ages 20 and 40, with a tapering pattern in older age groups.

Box Plots: These were used to visualize the spread and central tendency of numeric variables like driver age. The box plot highlighted the interquartile range and outliers, helping us identify unusually high or low values.

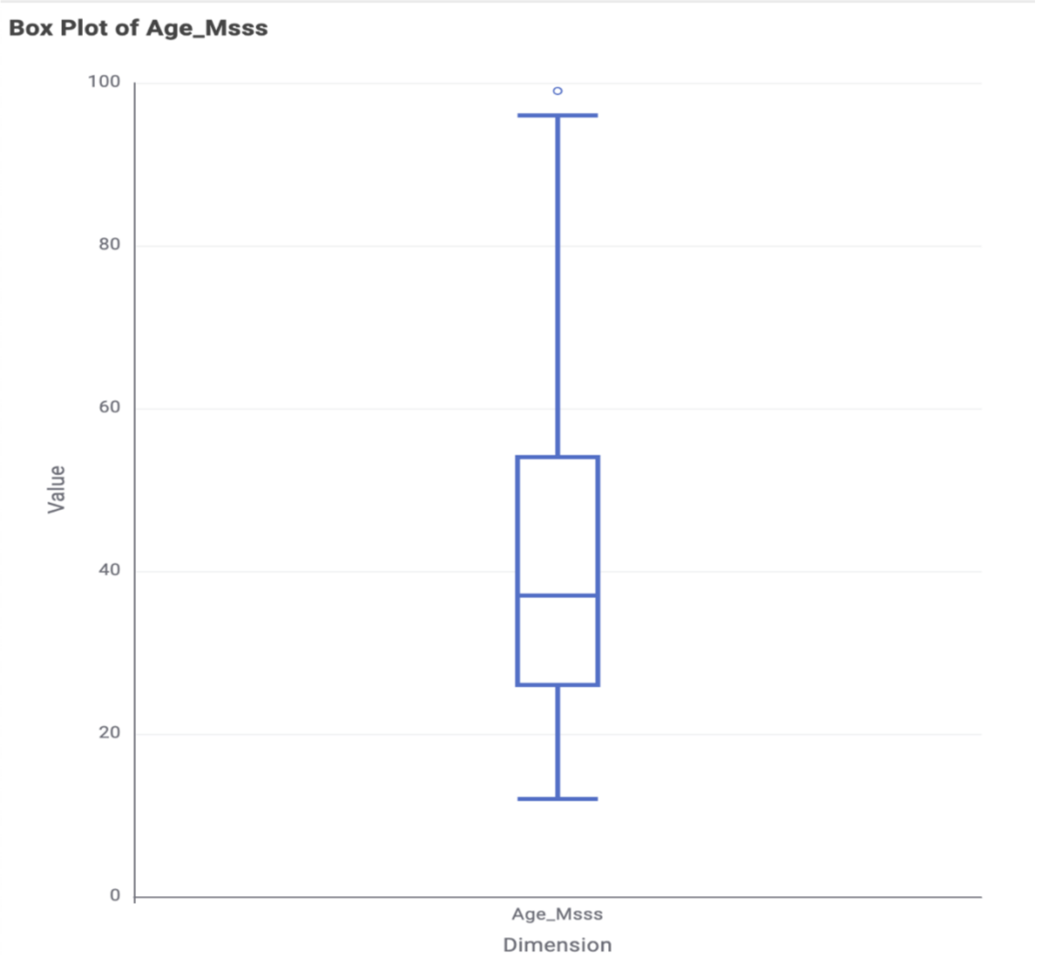


Figure 6: Box plot illustrating the central tendency and variability in driver ages. The median age is around 35, with a few high-end outliers nearing age 100.

Insights from the Data

Several insights emerged from this data exploration process. First, there is a pronounced class imbalance in the injury severity variable, with most individuals experiencing minor or no injuries. This will require techniques like SMOTE or stratified sampling to balance training data for supervised learning models.

Second, while many features show promise as predictors (e.g., restraint use, speed, age, vehicle maneuver), a subset of variables will need to be excluded due to excessive missing values or low variance. Third, some variables are highly correlated (like travel speed and posted speed limit), and dimensionality reduction techniques may be applied during modeling.

Lastly, the data's relational nature—multiple people and vehicles per crash—suggests that data merging must be performed carefully to avoid duplication or loss of granularity. We'll use CASENUM, VEH_NO, and PER_NO hierarchically to preserve relationships during integration.

This phase has laid a solid foundation for the upcoming Data Preparation phase, where we will clean, transform, and merge the data into a single training-ready dataset for machine learning tasks.

Data Preparation

Following the initial examination and quality checking of the CRSS datasets, data preparation focused systematically on transforming the intricate, real-world crash data into a clean, structured, and modeling-ready format. Since the original datasets were relational in nature accident-level, vehicle-level, person-level, and distraction-related records—the process of accurately joining the datasets on hierarchical keys (CASENUM, VEH_NO, PER_NO) was crucial to preserve the distinctive information added at each level.

Subsequently, targeted filtering operations were applied to restrict the dataset to single-driver passenger cars in which a valid injury severity rating was available. By narrowing the data in this way, not only was analytical focus maintained consistently, but the data were also aligned with proposed goals for predictive modeling.

In recognition of the considerable missing values, poorly coded data, and skewness among variables, considerable effort in data cleaning was made. Missing or ambiguous entries, particularly those with codes of -1, 99, or 999, were either imputed, recoded, or flagged using advanced KNIME nodes such as Math Formula, Rule Engine, and Missing Value. Engineered features were also created to extract more informative information from already available attributes, examples include calculation of vehicle age, over-speeding flags, and binning of continuous variables such as number of occupants, speed, and age.

In most of the encoded categorical fields, there was significant effort made to decode cryptic numerical codes into readable, human-understandable labels. This was done consistently through Rule Engine nodes, where all resulting categorical variables were cleanly differentiated by a "Str" suffix in their naming scheme. This framework enabled the use of adaptive encoding techniques in downstream modeling pipelines: numerical models like Logistic Regression and MLP required

One-Hot Encoding, while set-based models like Decision Trees and Random Forests utilized the original nominal columns without modification.

Finally, to counteract the bias found in the injury severity class sets, balancing techniques including SMOTE and Equal Size Sampling were utilized pragmatically depending on model form. Preprocessing resulted in a well-formed, adaptable dataset with open variable semantics, minimal missing values, and improved compatibility with multiple machine learning frameworks.

3.1 Filtering and Merging

Due to the relational CRSS data structure, the first necessary step toward preparing the dataset was a strict and integrity-constraining merge of the four source tables: accident-level, vehicle-level, person-level, and distraction-related records. Each table captured distinct, complementary details of crash events. To best make use of this multi-dimensional information, data sets were hierarchically connected on the main key CASENUM, with VEH_NO and PER_NO where necessary to ensure vehicle-specific and person-specific distinctions. This guaranteed each observation retained its original context within the crash event community.

With the incidence of many-to-one and one-to-many relations, like many people for a single vehicle, or many vehicles for an accident, precautionary join techniques came into practice. Inner joins were employed most frequently to retain cases that have complete corresponding records in both the datasets to avoid having partially incomplete merged records. Where there was no direct match for some datasets like the Distraction table, left joins were employed to keep primary crash and vehicle data even if distraction data was not present.

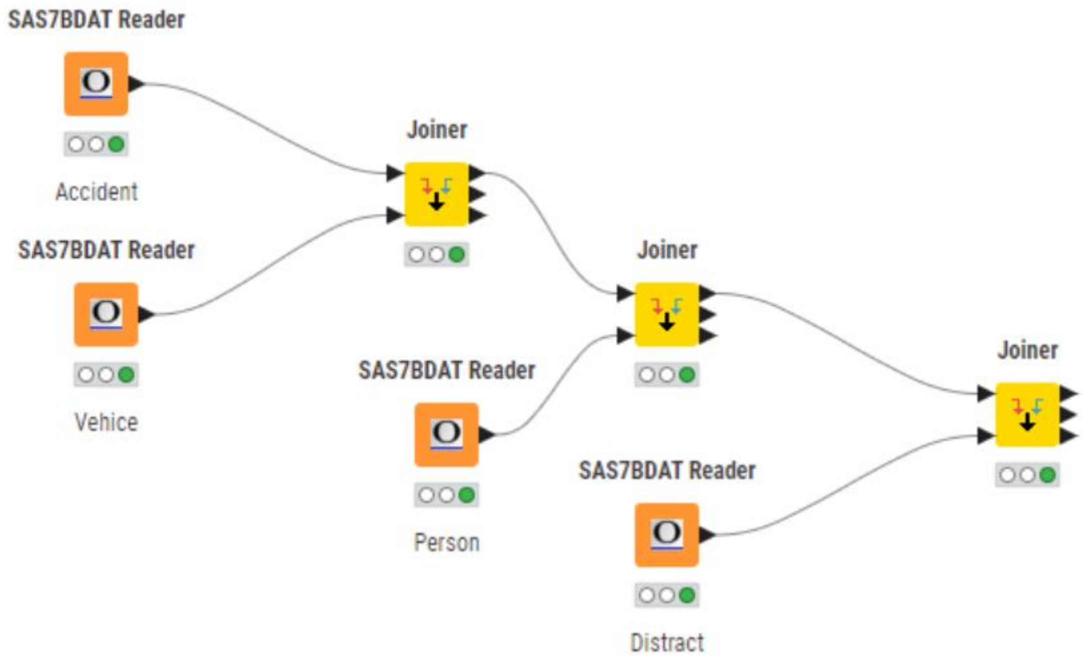


Figure 7: Merging the four CRSS datasets in KNIME using Joiner nodes and hierarchical keys

Following merging, an aggressive record filtering method was applied to align the dataset with the scope of predictive modeling that was intended. The following was employed:

Vehicle Type Limitation: Only passenger vehicles (sedans, SUVs, light trucks, vans) were maintained. Non-passenger vehicles such as motorcycles, heavy trucks, buses, and special vehicles were excluded to eliminate heterogeneity in crash dynamics.

Driver-Only Consideration: Data were conditioned to be limited to cars with drivers. Cars without drivers were excluded to keep it uniform.

Valid Injury Severity Range: Crashes with injury severity ratings ranging from 0 (possible injury) to 4 (incapacitating injury) were retained. “Died before crash” (level 6) and missing/unknown injury severity were removed to guarantee data quality and model suitability.

Deletion of Duplicates: To avoid data redundancy, merged key checks (CASENUM, VEH_NO, PER_NO) were performed to guarantee that no duplicate entities were present in the filtered dataset.

These selection filtering options made the final working population a stable, interpretable, and policy-relevant sample of crash-injured passenger car drivers. The selection criteria also weighed specificity and generalizability so that the resulting models would provide insights that are specific and applicable more generally.

During this step, a new binary injury severity variable (Injury_Sev_Binned) was constructed through a Numeric Binner node. This transformation reduced injury outcomes into two bins “Minor" (less severe injuries) and "Major" (severe injuries), simplifying the modeling task to a binary classification task but preserving critical distinctions between different crash outcomes.

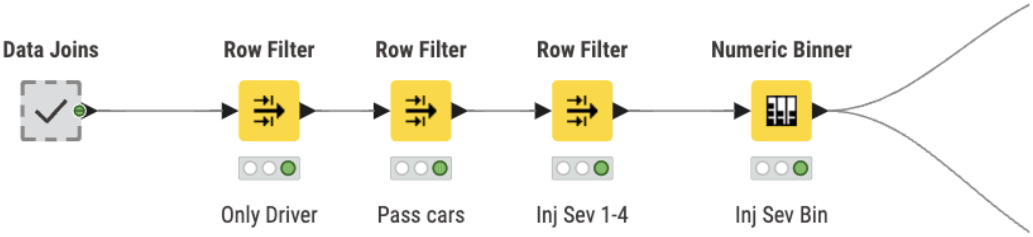


Figure 7: Filtering workflow in KNIME to isolate driver-only records from passenger cars with valid injury severity (1–4), followed by binning into Minor and Major injury classes.

This early combination and filtering process exhibited an immense reduction in dataset size but experienced an astronomical rise in analytical coherence so that further modeling work would have training on high-integrity, scenario-consistent crash records.

3.2 Missing Value Management

Handling missing or inconsistent data formed a significant subphase in data preparation because actual-world CRSS datasets heavily relied on post-crash reporting in jurisdictions and fields tended to have often substantial levels of incompleteness by underreporting, nonresponse, or through conditional nature of data capture (e.g., some fields only reported for serious crashes).To obtain a solid and clean dataset, we utilized a multi-strategy approach to missing values handling, using flagging, imputation, transformation, and selective deletion, and all of this within KNIME by a sequence of Missing Value, Math Formula, and Rule Engine nodes.

3.2.1 Flagging and Identification

The first task was to mark and label missing data clearly, especially for columns where missingness had been represented as special numeric codes (e.g., -1, 98, 99, 999). Although they were not null values in themselves, they semantically meant "Unknown" as well as "Not Applicable" or "Not Reported."

These codes were standardized and re-encoded as valid missing indicators in Rule Engine nodes. For instance, new binary flags like ModYear_Miss, Speed_Miss, and Age_Miss were added to mark missingness explicitly for later analysis and potential use as features.

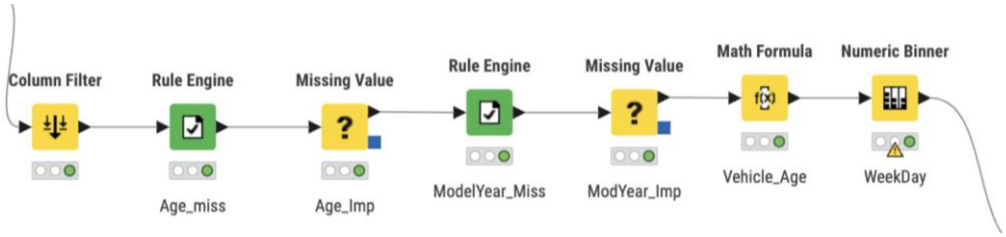


Figure 8: Imputation and transformation pipeline in KNIME for age and model year, including derivation of Vehicle Age and binning of crash day into weekday/weekend.

3.2.2 Imputation and Derivation

Second, imputation was selectively used to recreate reasonable estimates for important numerical attributes at the cost of information distortion. It was especially essential in modeling-critical variables where direct deletion would cause significant loss of data.

Critical imputations were:

1. **Vehicle Age (Vehicle_Age):** Computed by executing a subtraction between crash year and imputed model year with a Math Formula node. Imputation of model year was based on domain-relevant ranges (e.g., replacing 9999 with mode or median year).
2. **Overspeed Indicator:** Derived by comparing imputed travel speed to sign posted speed limit. In cases where both were absent, a conservative default (e.g., not over speeding) was used unless contextual hints suggested otherwise.

In cases where imputation was not applicable notably for nominal variables like airbag deployment or restraint use missing values were left as an "Unknown" category using Rule Engine logic. This ensured interpretability without the undue removal of records.

3.2.3 Targeted Removal

In certain high-impact modeling scenarios, rows with significant missing values were entirely removed — but only where they comprised a negligible proportion of the dataset (e.g., <5%) and were of high predictive importance. For example, rows with an invalid injury severity score were pre-emptively removed during the filtering process (see Section 3.1), as these would have an instantaneous detrimental impact on the modeling target. This selective pruning retained data integrity without diminishing sample size excessively. This fine-grained, column-wise management of missing values assured modeling inputs that were both statistically valid and semantically coherent. The result was a thoughtfully groomed dataset with essential attributes maintained, distortions minimized, and uncertainty clearly encoded.

3.3 Feature Engineering and Transformation

To enhance the predictive power of the dataset and tailor it to the machine learning models used in this study, extensive feature engineering was performed. This phase involved creating new variables, binning numeric attributes, categorizing coded values into interpretable groups, and ensuring consistency across variable types. The goal was to extract more meaningful patterns from raw crash data while ensuring compatibility with both numeric-based and set-based models.

Feature engineering was carried out in a modular pipeline in KNIME using a series of Rule Engine, Numeric Binner, Math Formula, Column Renamer, and Missing Value nodes. Each transformation was designed with domain knowledge, model compatibility, and interpretability in mind. A key convention was adopted: any feature name ending in “Str” refers to a string-type column created using Rule Engine nodes.

3.3.1 Numeric Transformations and Binning

Several numerical variables were transformed to highlight latent patterns or normalized scales for classification modeling. The following derivations and binning's were implemented:

Vehicle Age (Vehicle_Age): Computed by subtracting the imputed model year from the crash year. This helped capture the relative age of the vehicle, an important proxy for safety features and mechanical condition.

Overspeed Indicator (Overspeed): A binary flag derived by comparing actual travel speed with the posted speed limit. If the travel speed exceeded the limit, the record was flagged as over speeding.

Urbanicity (Urban Rural Bin): Binned from an underlying variable representing urban density or location type. Created using a Numeric Binner to distinguish urban, suburban, and rural zones.

Number of Vehicles (No of Veh Bin Str): Number of vehicles involved in the crash was binned to reduce noise and make interpretation easier for classification.

Weekday vs Weekend (Weekday): Derived from the crash day variable to differentiate weekday vs weekend crashes, acknowledging behavioral differences in traffic patterns.

Number of Occupants (Num Occup Bin Str): Transformed from a numeric count into labeled bins to identify single vs multi-occupant vehicles, which can impact injury likelihood.

Each of these binned variables was created using Numeric Binner nodes where appropriate and appended as new columns, allowing original values to be retained if needed.

3.3.2 Categorical Consolidation and Rule-Based Labeling

The CRSS dataset relied heavily on coded numeric representations for categorical variables (e.g., 1 = “Male”, 2 = “Female”, 98 = “Not Reported”), which were not model-ready. To make these variables interpretable and actionable, Rule Engine nodes were applied extensively. Each Rule Engine mapped codes to human-readable labels and produced a new string-type column, clearly named with the Str suffix for clarity.

Notable engineered string variables included:

- **Crash Conditions**
- **Light Condition Bin Str:** Daylight, dark-lighted, dark-not lighted, unknown
- **Weather Bin Str:** Clear, rainy, snowy, foggy, unknown
- **RoadAlign Str, Road Slope Str, Road Surf Cond Str:** Road curvature, elevation, and surface state
- **Driver Characteristics**
- **Sex Bin Str:** Male, Female, Unknown
- **Drugs Bin Str:** Drug involvement coded into Yes/No/Unknown
- **Ejection Bin Str, Restraint Bin Str, Airbag Bin Str:** Binary or ternary indicators of safety equipment use
- **Typ Intersection Bin Str:** Type of intersection - T, 4-way, roundabout, etc.

- **Crash Mechanics and Impact**
- **Rollover Cat:** Degree/type of rollover
- **Towed Cat:** Whether the vehicle was towed
- **Deformed:** Extent of vehicle damage
- **Fire (0/1):** Binary flag indicating whether a fire occurred during the crash
- **Environmental and Contextual Features**
- **WrkZone Str:** Construction zone involvement
- **TrafWay Str**: Roadway configuration (e.g., divided, one-way)

In cases where many codes existed for a feature (e.g., vehicle body type, crash configuration), less frequent codes were grouped under an "Other" or "Unknown" label to reduce dimensionality and prevent overfitting.

Variable Semantics and Model Compatibility:

At the end of the engineering pipeline, variables were reviewed for compatibility with different model architectures. Specifically, **Set-based models** (e.g., Decision Tree, Random Forest) used the raw Str columns directly, since these algorithms can natively process categorical data. **Numeric-based models** (e.g., Logistic Regression, MLP) required One-Hot Encoding. The "Str" columns were encoded using KNIME's "One to Many node", which exploded categorical columns into binary indicator columns.

As a result of this processing, the final feature set consisted of:

Numerical columns (e.g., Vehicle_Age, Overspeed)

Engineered string columns (e.g., Sex Bin Str, Light Condition Bin Str, Rollover Cat)

One-hot encoded columns for numeric models

This setup enabled flexibility and reusability across modeling workflows, while maintaining transparency and explainability in how each input feature was constructed.

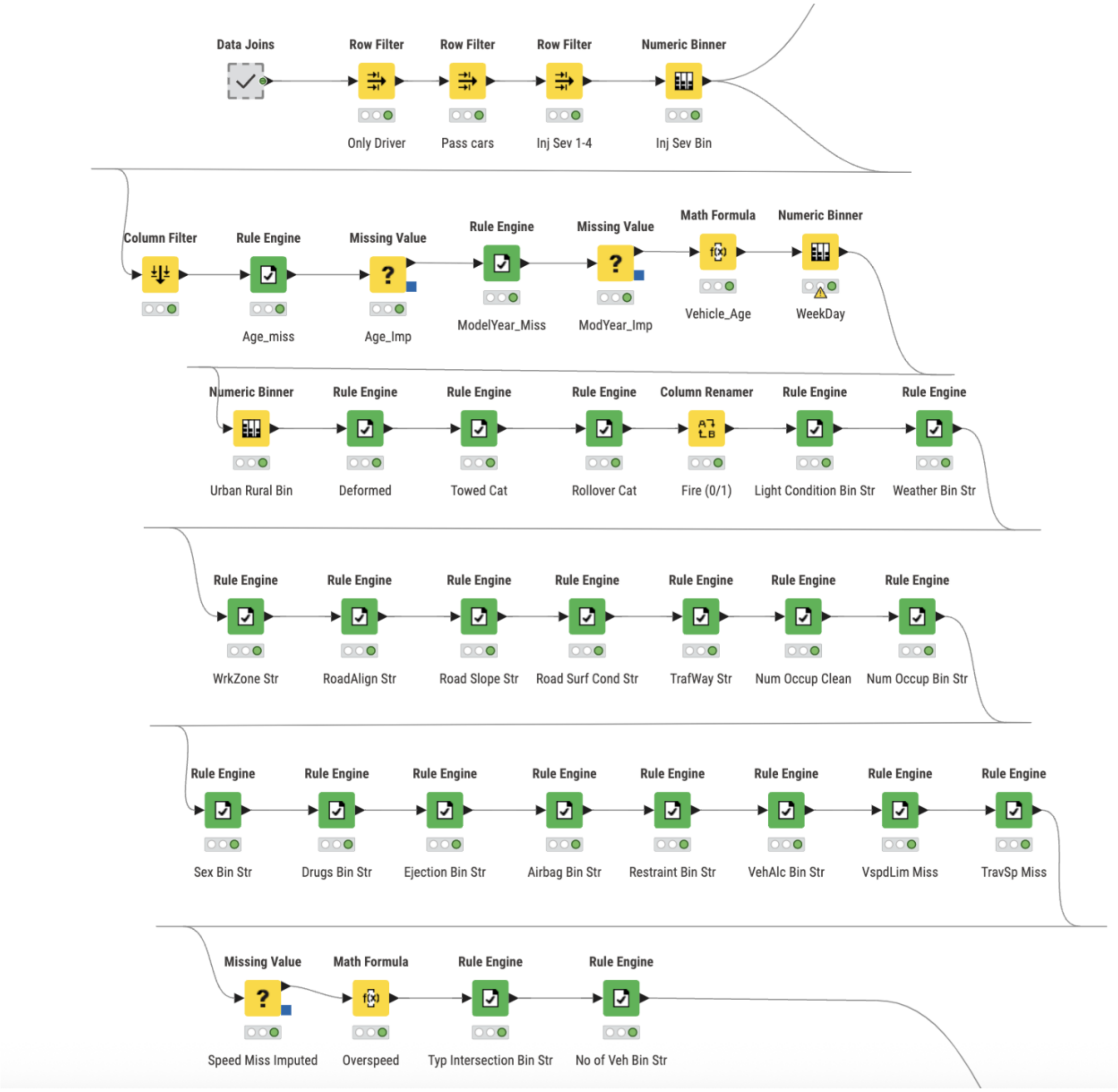


Figure 9: Feature engineering pipeline in KNIME transforming raw crash data into structured, model-ready variables.

3.4 Encoding and Class Balancing

After cleaning and feature set engineering, additional transformations were necessary to make the dataset fully appropriate for the machine learning algorithms selected for modeling. This included encoding categorical variables into numeric form where necessary and implementing a class balancing technique to adjust for extreme skewness in the distribution of injury severity.

3.4.1 Variable Encoding Strategy

The final engineered dataset included a combination of numeric variables, categorical string variables (with "Str" suffix), and generated binary indicators. Since various modeling techniques have various input requirements, two encoding paths were established:

Set-Based Models (e.g., Decision Tree, Random Forest):

These models inherently deal with nominal categorical variables. So, the engineered string variables (i.e., Sex Bin Str, Weather Bin Str) were directly fed into these models without one-hot encoding. This preserved semantic structure and avoided unnecessary dataset dimensionality increase.

Numeric-Based Models (e.g., Logistic Regression, Multi-Layer Perceptron):

For algorithms that accepted only numerical input, categorical variables were transformed using a One to Many (One-Hot Encoding) node in KNIME. This included binary indicator columns for each category level, adding a significant number of features but maintaining model compatibility.

It was taken care of to avoid perfect multicollinearity by optionally removing a reference category where appropriate, according to model assumptions.

This two-fold encoding strategy ensured that all the machine learning algorithms were trained with well-formatted data devoid of concealed biases or technical issues due to incompatibility of variable types.

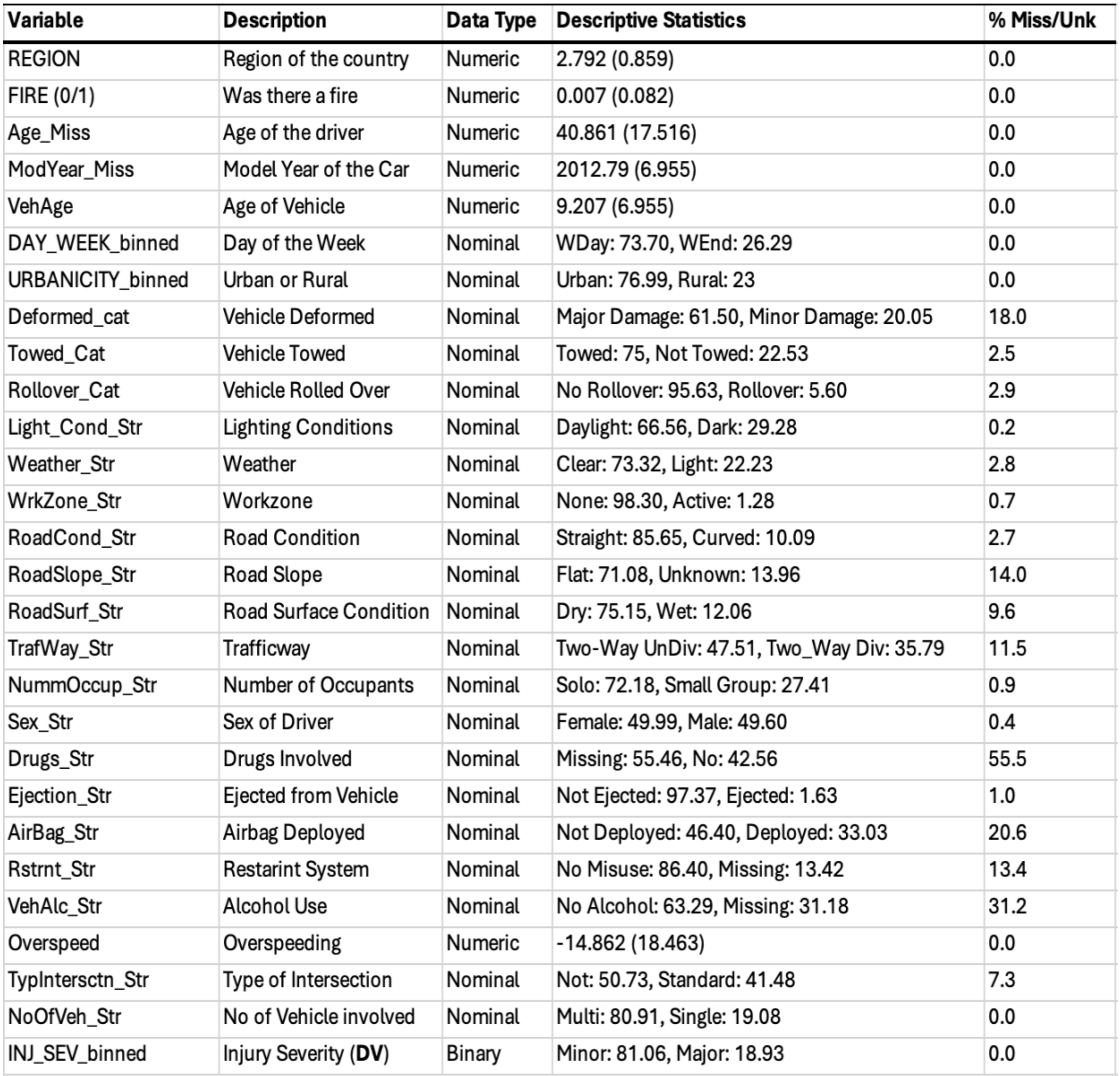
3.4.2 Class Imbalance Handling

Preliminary exploration revealed severe class imbalance in the binary target feature (INJ_SEV_binned), with minor injuries (Classes 1–2) significantly out-numbering major injuries (Classes 3–4). If not adjusted, this imbalance would result in model predictions biased towards the prevalent class, reducing the usefulness of injury severity risk predictions.

To correct this imbalance, Equal Size Sampling was employed in all modeling pipelines.

Using KNIME's Equal Size Sampling node, the data set was split so that both injury classes were represented by approximately equal numbers of records.

With the application of Equal Size Sampling, we could train the models on balanced datasets without introducing synthetic samples, thereby maintaining the natural feature distributions of the real-world crash data.



Variable	Description	Data Type	Descriptive Statistics	% Miss/Unk
REGION	Region of the country	Numeric	2.792 (0.859)	0.0
FIRE (0/1)	Was there a fire	Numeric	0.007 (0.082)	0.0
Age_Miss	Age of the driver	Numeric	40.861 (17.516)	0.0
ModYear_Miss	Model Year of the Car	Numeric	2012.79 (6.955)	0.0
VehAge	Age of Vehicle	Numeric	9.207 (6.955)	0.0
DAY_WEEK_binned	Day of the Week	Nominal	WDay: 73.70, WEnd: 26.29	0.0
URBANICITY_binned	Urban or Rural	Nominal	Urban: 76.99, Rural: 23	0.0
Deformed_cat	Vehicle Deformed	Nominal	Major Damage: 61.50, Minor Damage: 20.05	18.0
Towed_Cat	Vehicle Towed	Nominal	Towed: 75, Not Towed: 22.53	2.5
Rollover_Cat	Vehicle Rolled Over	Nominal	No Rollover: 95.63, Rollover: 5.60	2.9
Light_Cond_Str	Lighting Conditions	Nominal	Daylight: 66.56, Dark: 29.28	0.2
Weather_Str	Weather	Nominal	Clear: 73.32, Light: 22.23	2.8
WrkZone_Str	Workzone	Nominal	None: 98.30, Active: 1.28	0.7
RoadCond_Str	Road Condition	Nominal	Straight: 85.65, Curved: 10.09	2.7
RoadSlope_Str	Road Slope	Nominal	Flat: 71.08, Unknown: 13.96	14.0
RoadSurf_Str	Road Surface Condition	Nominal	Dry: 75.15, Wet: 12.06	9.6
TrafWay_Str	Trafficway	Nominal	Two-Way UnDiv: 47.51, Two_Way Div: 35.79	11.5
NummOccup_Str	Number of Occupants	Nominal	Solo: 72.18, Small Group: 27.41	0.9
Sex_Str	Sex of Driver	Nominal	Female: 49.99, Male: 49.60	0.4
Drugs_Str	Drugs Involved	Nominal	Missing: 55.46, No: 42.56	55.5
Ejection_Str	Ejected from Vehicle	Nominal	Not Ejected: 97.37, Ejected: 1.63	1.0
AirBag_Str	Airbag Deployed	Nominal	Not Deployed: 46.40, Deployed: 33.03	20.6
Rstrnt_Str	Restarint System	Nominal	No Misuse: 86.40, Missing: 13.42	13.4
VehAlc_Str	Alcohol Use	Nominal	No Alcohol: 63.29, Missing: 31.18	31.2
Overspeed	Overspeeding	Numeric	-14.862 (18.463)	0.0
TypIntersctn_Str	Type of Intersection	Nominal	Not: 50.73, Standard: 41.48	7.3
NoOfVeh_Str	No of Vehicle involved	Nominal	Multi: 80.91, Single: 19.08	0.0
INJ_SEV_binned	Injury Severity (**DV**)	Binary	Minor: 81.06, Major: 18.93	0.0

Table 2: Final variable list with simple statistics.

(For numeric variables: mean (St. Dev.); for binary or nominal variables: % frequency of the top two classes.)

Modeling

Modeling is an important phase of the machine learning algorithm where algorithms are trained from historical data to detect patterns and correlations between input features and the target variable. The goal is to develop a forecast model that can accurately make choices or predictions when applied to new, unseen data. This involves selecting the correct algorithms, training them on part of the data, and validating them using the correct metrics. Successful modeling interprets raw data into useful insights, and therefore a crucial step towards solving real-world problems through data-driven solutions.

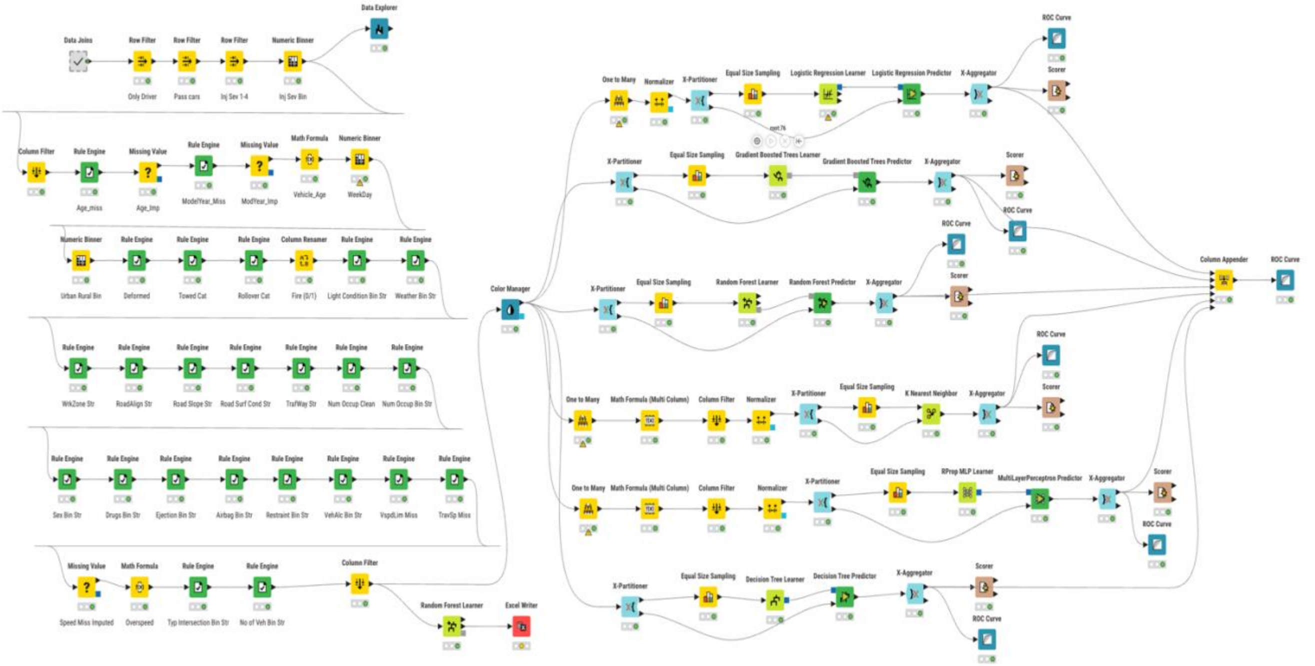


Figure 10: Integrated modeling workflow in KNIME showing preprocessing, one-hot encoding, equal size sampling, and parallel execution of classification models with ROC-based evaluation.

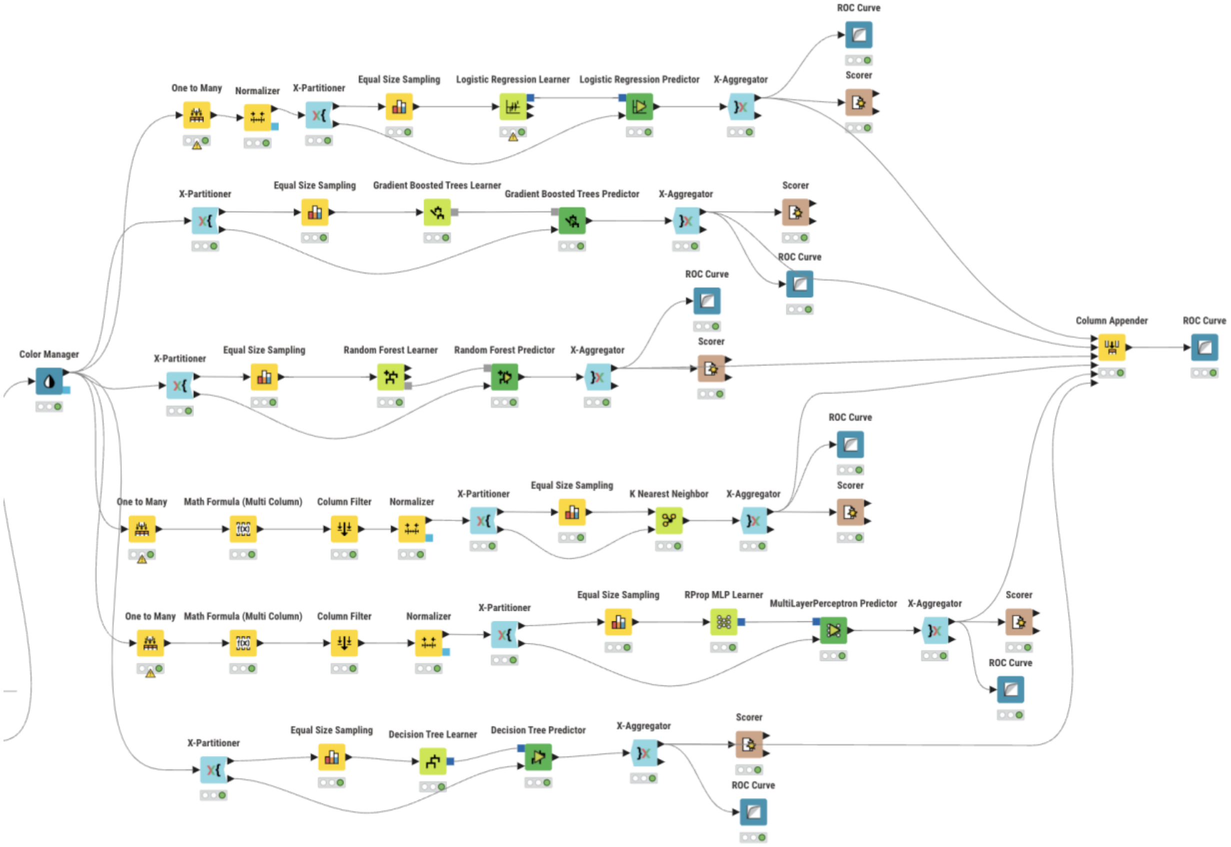


Figure 11: Modeling workflow close-up showing five machine learning pipelines (LR, GBT, RF, DT, MLP) developed with balanced sampling and evaluated through ROC curve comparison.

Logistic Regression Model:

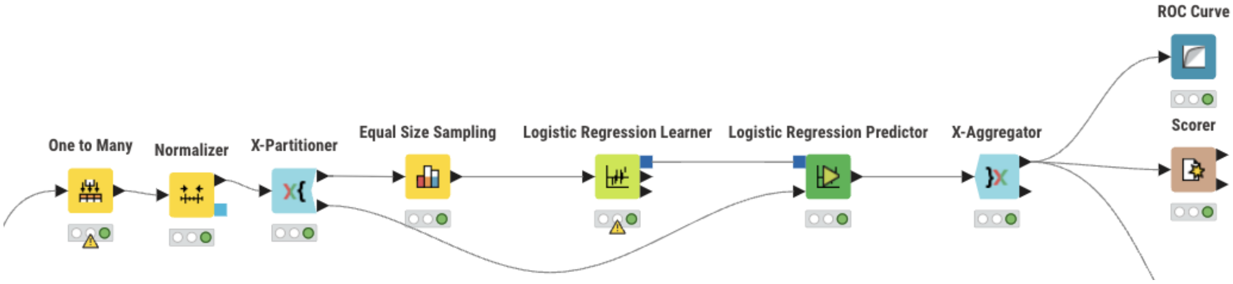


Figure 12: Workflow of the Logistic Regression Model

The model's performance in the class gave it 31,677 true positives and 5,250 false positives, along with 13,686 true negatives and 1,256 false negatives. It had a recall of 0.716, which meant that it correctly classified 71.6% of the positive cases, while its precision was much lower at 0.376, where only 37.6% of its predicted positives were. This indicated a very high salient number of false positives. The recall, also 0.716, is consistent with the sensitivity, and specificity of 0.723 means the model correctly labeled 72.3% of non-positive instances. The F-measure, on average between recall and precision, was 0.493, showing moderate performance but capturing the precision-recall trade-off. Overall, the model was correct at 0.721, correctly labeling 72.1% of all instances, and Cohen's Kappa of 0.326, showing moderate agreement above chance. These results suggest that the model is quite good at detecting positive cases but is plagued by a high false positive rate, which could impact its applied reliability. Addressing the problem of precision, possibly with sophisticated feature selection or thresholding, could improve its overall effectiveness for this class.

Gradient Boosting Model:

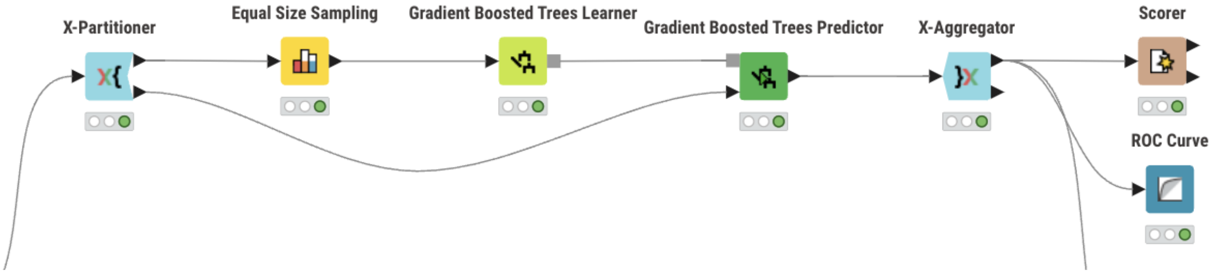

X-Partitioner
Equal Size Sampling
Gradient Boosted Trees Learner
Gradient Boosted Trees Predictor
X-Aggregator
Scorer
ROC Curve

Figure 13: Workflow of the Gradient Boosting Model

The model's performance was evaluated using a pipeline that included X-Partitioner for data splitting, Equal Size Sampling for balancing, Gradient Boosted Trees for classification, X-Aggregator for result aggregation, and a Scorer with an ROC Curve. Results showed 31,784 true positives, 5,156 false positives, 13,780 true negatives, and 1,239 false negatives. The model achieved a recall (sensitivity) of 0.720, correctly identifying 72% of positive instances, but had a precision of only 0.382, reflecting a high false positive rate. Specificity was 0.728, meaning 72.8% of negatives were correctly classified. The F-measure was 0.499, indicating moderate performance and a clear precision-recall trade-off. Overall accuracy was 0.726, with a Cohen's Kappa of 0.334, suggesting moderate agreement beyond chance. These findings highlight the model's reasonable ability to detect positives but suggest that tuning the Gradient Boosted Trees algorithm or adjusting the classification threshold could improve precision and reliability.

Random Forest Model:

Its performance on the targeted class was evaluated through a pipeline containing X-Partitioner, Equal Size Sampling, Random Forest Learner and Predictor, X-Aggregator, and a Scorer with ROC Curve. It achieved 30,111 true positives, 4,881 false positives, 14,055 true negatives, and 1,412 false negatives and achieved a recall of 0.681, precision of 0.382, sensitivity of 0.681, specificity of 0.742, and an F-measure of 0.489. Overall accuracy was 0.731, and Cohen's Kappa was 0.325, indicating moderate agreement.

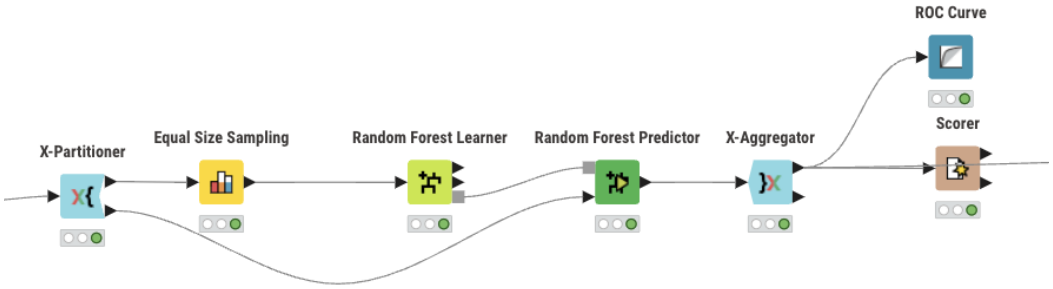


Figure 14: Workflow of the Random Forest Model

The model has strong positive detection but strong false positive detection, which could be improved with hyperparameter tuning or adjustment to sampling to raise precision.

Feature Importance Ranking from Random Forest Model:

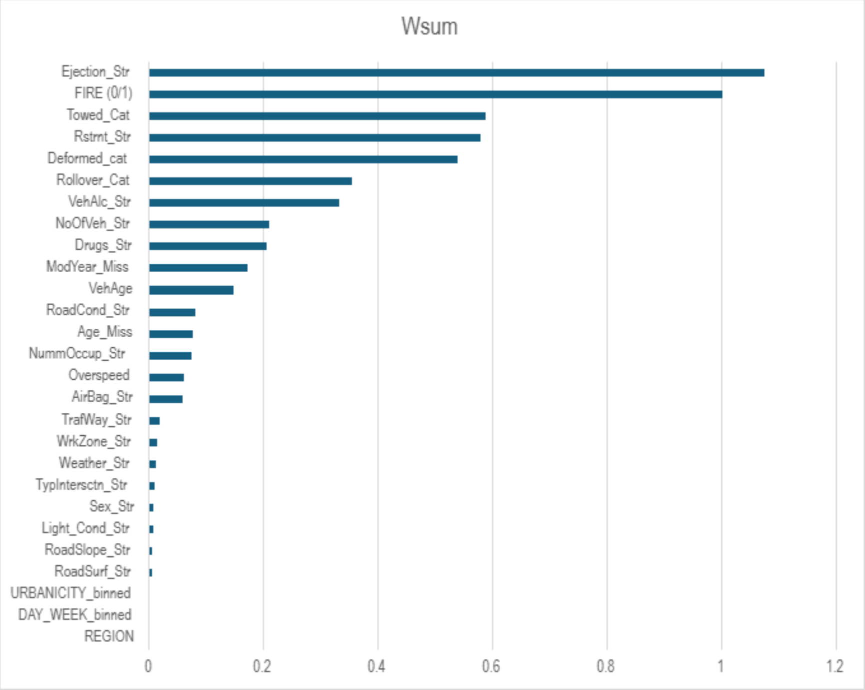


Figure 15:Variable importance chart showing the top predictors of injury severity based on model-driven feature contributions.

Variable importance analysis identifies the most influential features in the model based on splits, candidate occurrences at decision tree levels, and weighted sum (Wsum). Ejection_Str has a Wsum of 1.076, with splits at level 0 being 21, level 1 being 26, and level 2 being 38, and then FIRE (0/1) with a Wsum of 1.003, having 18, 25, and 27 splits at levels 0, 1, and 2, respectively. Towed_Cat and Rstrnt_Str also lead with Wsums of 0.589 and 0.580, the latter having splits at levels of 11, 23, and 33. Deformed_cat (Wsum 0.539), Rollover_Cat (Wsum 0.354), and VehAlc_Str (Wsum 0.333) are important, while REGION, DAY_WEEK_binned, and URBANICITY_binned have a Wsum of 0, indicating they have minimal influence. Crash severity and occupant safety features are the dominant ones, suggesting that they are the most significant predictors, as opposed to demographic and environmental features. This can guide feature selection for model improvement.

K Nearest Neighbor (KNN) Model:

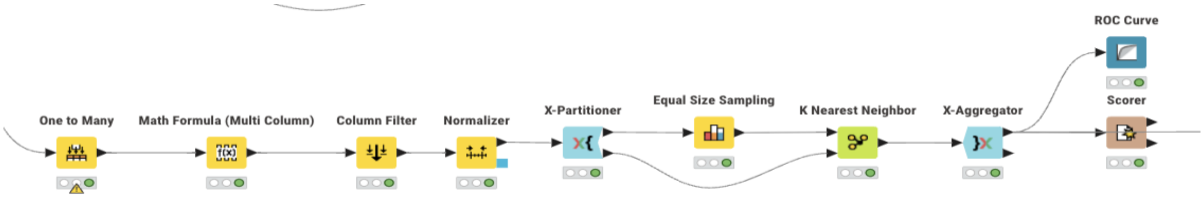


Figure 16: Workflow of the KNN Model

The K Nearest Neighbor (KNN) model was evaluated using a pipeline with data encoding, feature calculation, normalization, balancing, and scoring steps. Results showed 28,220 true positives, 6,034 false positives, 12,902 true negatives, and 1,603 false negatives, with recall and sensitivity at 0.638, precision at 0.319, specificity at 0.681, and an F-measure of 0.425. Overall accuracy was 67.3%, and Cohen's Kappa was 0.23, indicating modest agreement beyond chance. While the model shows modest success in identifying positives, high false positives suggest improvements through adjusting the number of neighbors (k) or refining feature selection.

Artificial Neural Networks Model (MLP):

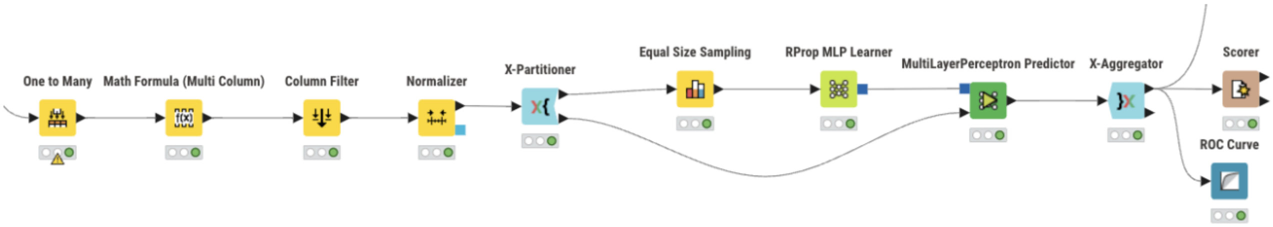


Figure 17: Workflow of the MLP Model

The model's performance on the target class was evaluated with a pipeline that included One to Many encoding, Math Formula (Multi Column) as a feature calculator, Column Filter and Normalizer for data preprocessing, X-Partitioner for data splitting, Equal Size Sampling as a method of sample balancing, RPProp MLP Learner and Multilayer Perceptron Predictor for classification, X-Aggregator to collect predictions, and a Scorer with an ROC Curve for performance checking. The results were 31,164 true positives, 5,502 false positives, 13,434 true negatives, and 1,259 false negatives, with recall of 0.715, precision of 0.365, sensitivity of 0.715, specificity of 0.709, and an F-measure of 0.483. The accuracy was 0.711, with a Cohen's Kappa of 0.311, indicating moderate agreement greater than chance. The Multilayer Perceptron model does quite well at identifying positive cases but has an extremely high false positive rate, which suggests that it might become better with changes to the structure of the neural network, such as how many layers or neurons, or adjusting the learning rate to enhance precision and overall reliability.

Decision Tree Model:

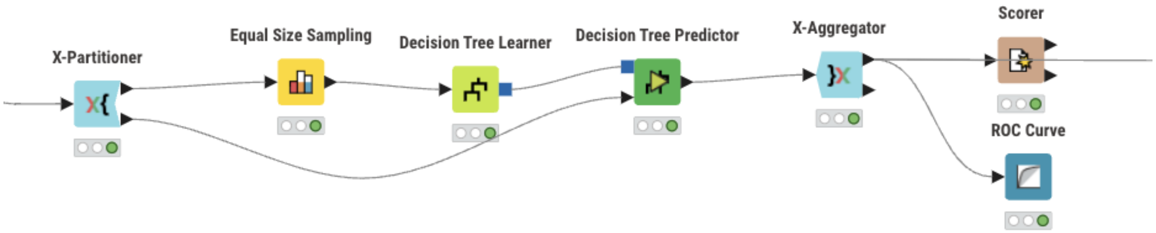
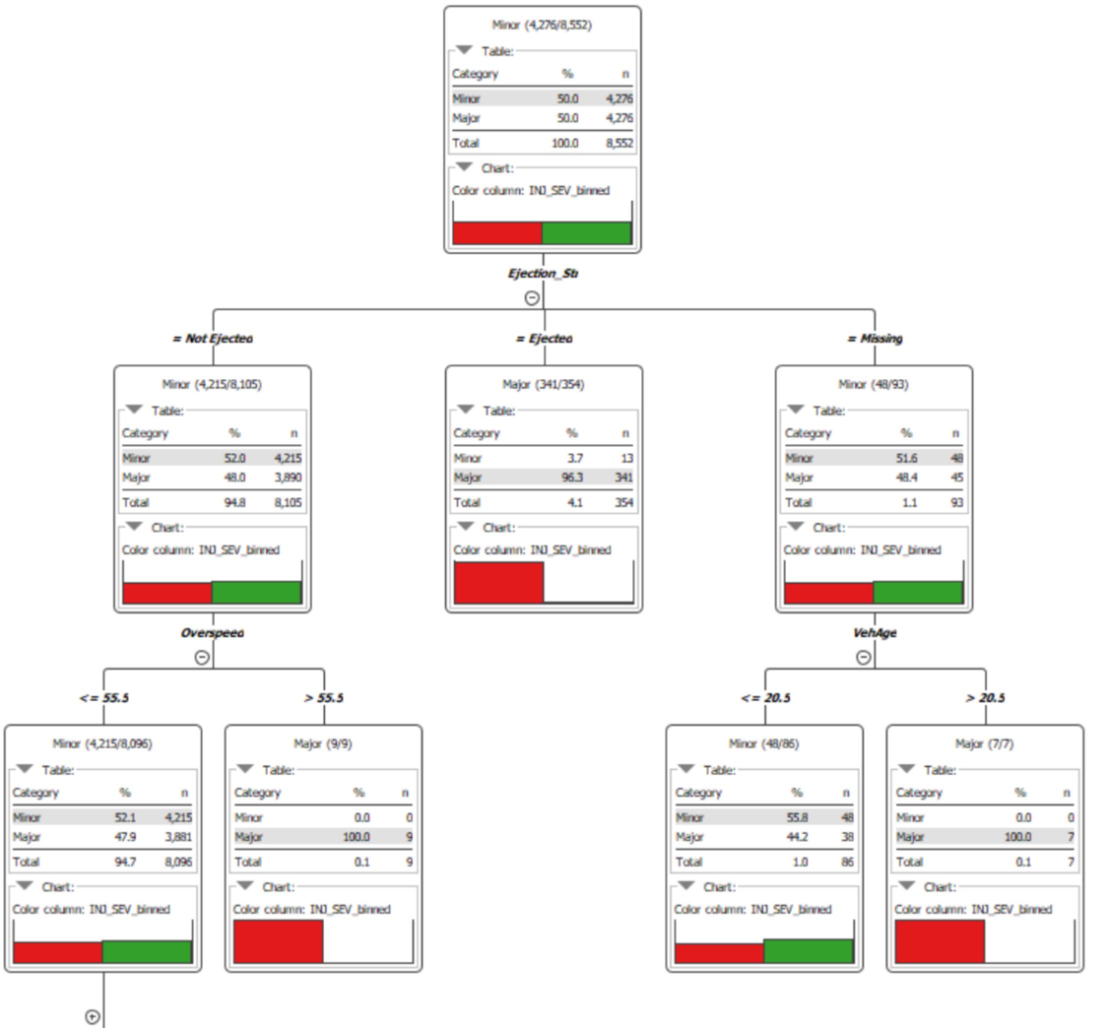


Figure 18:Workflow of the Decision Tree Model

The model's performance on the target class was evaluated using a pipeline consisting of X-Partitioner for data split, Equal Size Sampling to sample the data evenly, a Decision Tree Learner and Predictor to classify, X-Aggregator to predict aggregation, and a Scorer with ROC Curve to evaluate. The output showed 30,112 true positives, 5,422 false positives, 13,514 true negatives, and 1,411 false negatives, with recall being 0.681, precision 0.357, sensitivity 0.681, specificity 0.714, and an F-measure of 0.469. The overall accuracy was 0.707, with a Cohen's Kappa of 0.293, indicating modest agreement over chance. Decision Tree model performs well to an intermediate degree for picking up positive cases but shows high false positives, so it would be a priority to consider pruning the tree or adjusting split parameters to increase precision and general performance.

Three levels of the decision tree graphical model:

Decision tree classification (Fig. 19) illustrates the impact of significant variables on injury seriousness (Minor vs. Major). The first Ejection_Str split indicates that ejected occupants (4.1% cases) will have a 96.3% chance for major injuries while non-ejected occupants (94.8%) will have a more balanced 52% Minor and 48% Major split. In non-ejected scenarios, Overspeed also differentiates outcomes more: speeds ≤ 55.5 mph maintain a 52.1% Minor rate, and speeds > 55.5 mph get 100% Major injuries. In lower speeds, VehAge differentiates outcomes more, with older vehicles (VehAge > 20.5 years) still getting a 100% Major injury rate. These trends, as noted in Figure 19, identify ejection, overspeeding, and vehicle age as the primary predictors of severe injury outcome.



Minor (4,276/8,552)

Category	%	n
Minor	50.0	4,276
Major	50.0	4,276
Total	100.0	8,552

Color column: INJ_SEV_binned

Minor (4,215/8,105)

Category	%	n
Minor	52.0	4,215
Major	48.0	3,890
Total	94.8	8,105

Color column: INJ_SEV_binned

Major (341/354)

Category	%	n
Minor	3.7	13
Major	96.3	341
Total	4.1	354

Color column: INJ_SEV_binned

Minor (48/93)

Category	%	n
Minor	51.6	48
Major	48.4	45
Total	1.1	93

Color column: INJ_SEV_binned

Minor (4,215/8,096)

Category	%	n
Minor	52.1	4,215
Major	47.9	3,881
Total	94.7	8,096

Color column: INJ_SEV_binned

Major (9/9)

Category	%	n
Minor	0.0	0
Major	100.0	9
Total	0.1	9

Color column: INJ_SEV_binned

Minor (48/86)

Category	%	n
Minor	55.8	48
Major	44.2	38
Total	1.0	86

Color column: INJ_SEV_binned

Major (7/7)

Category	%	n
Minor	0.0	0
Major	100.0	7
Total	0.1	7

Color column: INJ_SEV_binned

Figure 19: Levels of the Decision Tree Model

Evaluation

In the evaluation phase, multiple machine learning models are assessed for their ability to predict injury severity (low vs. high) among drivers involved in passenger vehicle crashes. Each model is evaluated using performance metrics including precision, recall (sensitivity), specificity, and accuracy, with a particular emphasis on identifying high-severity injury outcomes (positive class) as shown in the Table 3 below:

Model	Sensitivity	Specificity	Precision	Accuracy
Logistic Regression	0.716	0.723	0.376	0.721
Gradient Boosting	0.720	0.728	0.382	0.726
Random Forest	0.681	0.742	0.382	0.731
K-Nearest Neighbors	0.638	0.681	0.319	0.673
Multilayer Perceptron	0.715	0.709	0.365	0.711
Decision Tree	0.681	0.714	0.357	0.707

Table 3: This table presents a comparative summary of each model's sensitivity, specificity, precision, and accuracy providing a comprehensive evaluation of their effectiveness in predicting injury severity outcomes.

The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) is computed to further evaluate each model's ability to distinguish between low and high injury severity outcomes, independent of any specific classification threshold. The AUC scores align with the model rankings observed through recall, and precision, as illustrated in the chart (Figure 19) below:

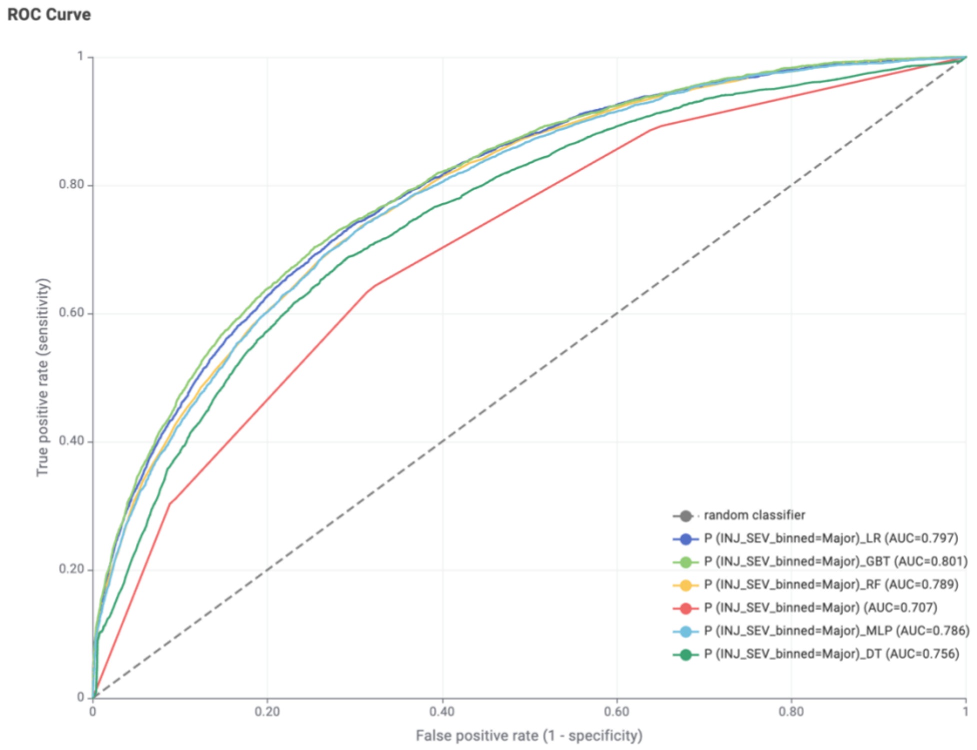


Figure 20: ROC curves for major injury severity prediction models. GBT achieved the highest AUC (0.801), while DT performed lowest (AUC = 0.756).

Logistic Regression

The logistic regression model achieves a sensitivity (recall) of 0.716 and a specificity of 0.723, indicating a strong balance in detecting both positive and negative injury severity classes. However, its precision is relatively low at 0.376, suggesting a high number of false positives among predicted severe cases. The model attains an overall accuracy of 72.1%, reflecting moderate agreement beyond chance. In terms of overall discriminative ability, the logistic regression model achieves an AUC of 0.797, confirming its strong capacity to distinguish between low and high injury severity outcomes across different threshold settings. While the model is effective at identifying serious cases and demonstrates robust class separability, its practical reliability is limited by the relatively low precision, suggesting a need for threshold optimization.

Gradient Boosting

Gradient Boosting demonstrates the best overall performance, achieving a sensitivity of 0.720 and a specificity of 0.728, indicating strong classification of both positive and negative injury severity outcomes. The model attains a precision of 0.382 and an overall accuracy of 72.6, reflecting moderate agreement beyond chance. Furthermore, Gradient Boosting records the highest AUC score of 0.801 among all models evaluated, confirming its superior discriminative ability across varying classification thresholds. These results position Gradient Boosting as the most reliable and balanced model for predicting driver injury severity in this study.

Random Forest

Random Forest delivers strong performance, achieving a sensitivity of 0.681 and the highest specificity among all models at 0.742, making it particularly effective at correctly identifying low-severity injury outcomes. The model matches Gradient Boosting in precision at 0.382. It attains an overall accuracy of 73.1%, indicating moderate agreement beyond chance. In terms of discriminative ability, Random Forest achieves an AUC of 0.789, confirming its strong capacity to differentiate between low and high injury severity levels across different threshold settings. Feature importance analysis highlights ejection status, fire occurrence, towed status, and restraint use as the most influential predictors, underscoring the dominant role of vehicle-related and crash-specific factors over demographic characteristics in determining injury severity.

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) model achieves the lowest sensitivity (0.638) and specificity (0.681) among all models evaluated, indicating limited ability to correctly classify both high- and low-severity injury outcomes. It attains a precision of 0.319 and an overall accuracy of 67.3%, reflecting only slight agreement beyond chance. In terms of discriminative ability, the KNN model records an AUC of 0.736, further confirming its relatively weak performance compared to other classifiers. These results suggest that KNN struggles to effectively distinguish between injury severity levels and would require significant optimization—such as adjusting the number of neighbors (k), implementing advanced normalization techniques, or applying feature engineering strategies—to achieve improved predictive accuracy.

Multilayer Perceptron (MLP)

The neural network model (MLP) records a sensitivity of 0.715 and a specificity of 0.709, indicating reasonably balanced classification between positive and negative injury severity outcomes. It achieves a precision of 0.365 and overall accuracy of 71.1, suggesting moderate agreement beyond chance. In terms of discriminative ability, the model attains an AUC of 0.786, which is comparable to the performance of logistic regression and confirms its solid, though not top-tier, separation between severity classes.

Decision Tree

The Decision Tree model achieves a sensitivity of 0.681 and a specificity of 0.714, demonstrating moderate ability to classify both high- and low-severity injury outcomes. It records a precision of 0.357 and an overall accuracy of 70.7%, indicating modest agreement beyond chance. In terms of discriminative power, the Decision Tree attains an AUC of 0.756, the lowest among the models evaluated, reflecting limited ability to consistently separate injury severity classes across threshold variations. Decision paths primarily rely on ejection status, overspeed, and vehicle age, reinforcing the significance of crash-related characteristics over demographic factors in determining injury outcomes.

Overall, all models demonstrate moderate success in detecting high-severity injury outcomes. Gradient Boosting and Random Forest emerge as the most reliable models when considering a combination of accuracy, F-measure, sensitivity, specificity, and AUC, offering the strongest balance between detection capability and generalization.

Deployment

The deployment phase focuses on applying the results of the modeling process in real-world settings to improve road safety outcomes and support data-driven decision-making. With the models trained, tested, and evaluated—particularly Gradient Boosting and Random Forest, which performed most reliably—the next step involves identifying the most effective ways to integrate these insights into operational systems, planning tools, and policy frameworks.

To support this transition from development to deployment, a complete workflow was designed and documented, with a full screenshot below (Figure 21) to demonstrate the end-to-end modeling process. This workflow visually captures each phase of the pipeline, from data preparation and sampling to model training, validation, and evaluation, providing a clear blueprint for operationalization.

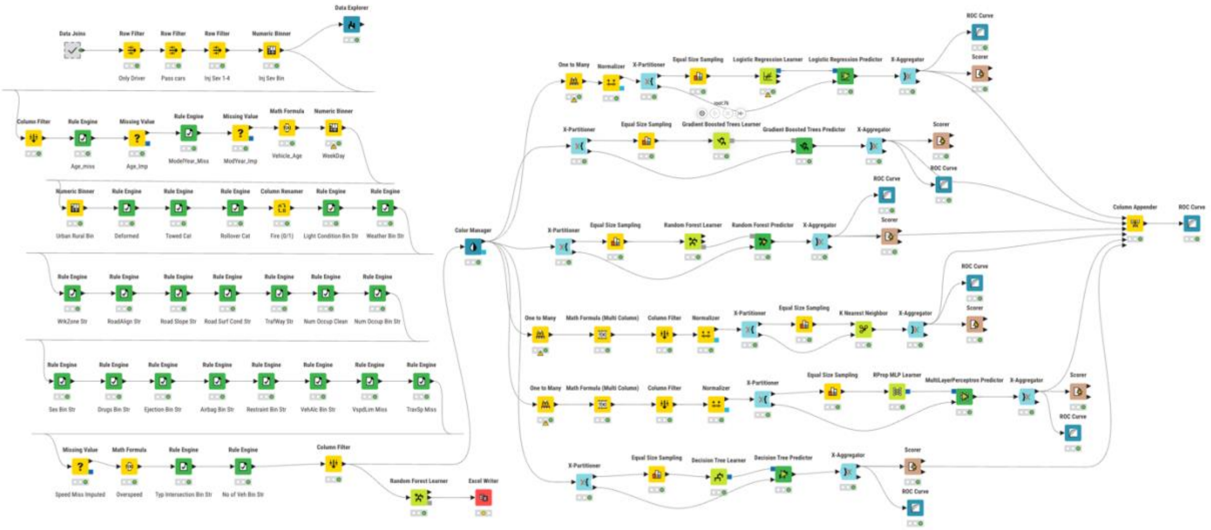


Figure 21: This figure presents the complete workflow from initial data preparation through model training, validation, and performance evaluation, outlining the entire predictive modeling pipeline used in the study.

Operational Integration and Use Cases

The predictive models developed in this study can be deployed across a variety of operational contexts. Government agencies such as the National Highway Traffic Safety Administration (NHTSA) or state Departments of Transportation can use the models to assess injury severity risk in crash analytics platforms. First responders and law enforcement can also benefit from these models by incorporating them into crash report systems, enabling quick, data-driven estimation of injury severity at the scene of an accident. Additionally, insurance companies could use predictions to enhance their risk profiling, claims evaluation, and fraud detection processes.

These applications can be supported through real-time integration into crash reporting interfaces or vehicle telematics systems, allowing instant risk assessment using variables like vehicle speed, restraint usage, vehicle condition, and crash configuration. A dashboard-based interface can present the likelihood of a major injury based on model output, helping guide immediate or long-term interventions.

Policy Development and Planning Support

Beyond immediate operational use, model insights can inform public safety policies and long-term transportation planning. Variables identified as the most influential—such as ejection status, restraint use, fire involvement, and vehicle age—provide a clear signal for targeted public awareness campaigns and educational programs. For instance, high risk associated with non-use of restraints supports the continued promotion of seatbelt enforcement programs.

From a planning perspective, areas with a high prevalence of severe crashes, identified through geospatial trends or environmental conditions in the data, can be prioritized for infrastructure upgrades or traffic regulation changes. Policymakers may also consider using model outputs to guide the formulation of vehicle safety regulations or to incentivize the adoption of crash-preventative technology in older or high-risk vehicles.

Technical Deployment Options

For practical implementation, the models can be deployed using platforms such as KNIME Server or KNIME WebPortal, offering a user-friendly interface for scoring new data.

In any technical deployment, it is important to maintain a pipeline for periodic model updates. As new crash data become available—particularly from sources like the Crash Report Sampling System (CRSS)—the models can be retrained to maintain relevance and predictive power over time. Continuous evaluation of model performance and alert systems for drift detection can help preserve accuracy and reliability.

Ethical and Practical Considerations

While deploying predictive models in traffic safety brings significant value, it also introduces several considerations. One key challenge identified in the evaluation phase is the low precision of most models, which implies a high number of false positives. This issue must be addressed before full-scale implementation, either through threshold calibration or the use of cost-sensitive learning techniques that penalize false positive predictions more heavily.

Additionally, privacy and ethics must be considered when integrating predictive analytics into systems that use personal or sensitive data. All models should be deployed with appropriate data protection measures, and personal identifiers should be removed or anonymized in compliance with privacy regulations such as HIPAA or GDPR. Transparency in model use and interpretability are also important, particularly when decisions based on model outputs may affect resource allocation or public services.

Next Steps for Deployment

To move forward with deployment, several next steps are recommended. First, a pilot implementation of the model should be conducted in a controlled environment, such as within a single agency or region, to assess usability and real-world performance. Feedback from users—including analysts, policymakers, and emergency responders—can help refine both the model and the interface.

Second, stakeholder collaboration will be essential to successful deployment. Engaging experts from transportation safety, public health, and software engineering will ensure that the model is well-integrated into existing workflows and systems. Clear documentation and training resources should also be developed to guide end users in interpreting and applying the model results correctly.

By following these steps, the project moves beyond analysis and into impact—empowering stakeholders with predictive tools that can guide proactive strategies to reduce severe injuries and fatalities on the road.

Summary

This project applies the CRISP-DM framework to develop and evaluate classification models for predicting injury severity in passenger vehicle crashes, using multi-source, real-world data from the Crash Report Sampling System (CRSS). The primary objective was to identify statistically significant and operationally meaningful predictors of injury severity, and to construct supervised learning models capable of distinguishing between minor and major injury outcomes at the individual driver level.

The analysis begins with a problem definition grounded in the public health and economic burden of traffic crashes in the United States. Despite advances in vehicle design, road engineering, and policy interventions, severe injuries remain prevalent, justifying the need for predictive models to support early risk assessment and strategic prevention efforts.

In the Data Understanding phase, four relational datasets—accident, vehicle, person, and distraction—were profiled to assess structure, distribution, and integrity. Hierarchical keys (CASENUM, VEH_NO, PER_NO) were used to preserve relationships across entity levels. Exploratory data analysis (EDA) revealed non-normal distributions in numeric features, dominant categories in environmental variables, and non-random patterns of missingness. Common data quality issues such as mixed missing value encodings (e.g., -1, 99, 999) and high-missingness variables were catalogued for harmonization in the preprocessing stage.

The Data Preparation phase involved systematic handling of missing data through deletion of high-missingness attributes and standardization of null representations. Redundant or low-variance variables were excluded, and categorical variables were decoded and recoded for interpretability. The data was filtered to include only driver-level observations from injury-causing passenger vehicle crashes. Given the class imbalance in the target variable (INJ_SEV_binned), an Equal Size Sampling approach was used to balance the minority and majority classes prior to model training.

Multiple classification models were developed in the Modeling phase, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Trees, K-Nearest Neighbors, and a Multilayer Perceptron (ANN). Each model was implemented within a KNIME-based pipeline, which included preprocessing, balanced sampling, k-fold cross-validation, and performance aggregation using the X-Aggregator node. Model performance was evaluated using class-specific metrics: recall (sensitivity), specificity, precision, F1-score, overall accuracy, Cohen's Kappa, and AUC.

The Evaluation phase demonstrated that Gradient Boosted Trees yielded the highest AUC (0.801), with strong recall (0.720) and specificity (0.728), indicating robust generalization capacity. Random Forest and Logistic Regression models also performed well, with AUC scores of 0.789

and 0.797, respectively. The Decision Tree and KNN models lagged in performance, with lower sensitivity and Kappa scores. Variable importance analysis highlighted ejection status, fire exposure, towed status, restraint use, vehicle damage, and vehicle age as top predictors—reinforcing the dominant influence of crash dynamics and occupant protection features over demographic variables.

In the Deployment phase, technical pathways were outlined for operationalizing model outputs. These include integrating predictive scores into crash report systems, emergency dispatch decision tools, and insurance claim triage processes. Deployment via KNIME Web Portal is recommended. Recommendations also include threshold optimization, retraining schedules, and model drift detection to ensure sustainable performance in dynamic data environments.

In summary, this project provides a data-driven framework for predicting injury severity using scalable machine learning methods applied to structured crash data. It offers both methodological rigor and practical insights, supporting its application in domains such as traffic safety analysis, policy evaluation, and injury prevention planning.

Conclusion

This study demonstrates the application of supervised machine learning to a structured, real-world crash dataset to predict injury severity among drivers in passenger vehicle collisions. Using the CRISP-DM methodology, we systematically progressed from business understanding to deployment planning, transforming raw CRSS data into a predictive modeling pipeline capable of generating actionable safety insights.

Through extensive data cleaning, integration, and feature selection, we prepared a dataset suitable for high-quality model training. Class imbalance was addressed using equal size sampling, and categorical variables were decoded to enhance interpretability. Multiple classification algorithms were evaluated, with Gradient Boosted Trees emerging as the most consistent performer across key metrics, including AUC, recall, and specificity. Logistic Regression and Random Forest also provided strong performance, balancing accuracy and interpretability.

Feature importance analysis revealed that injury severity is primarily influenced by crash-related and vehicle-specific factors—such as ejection status, restraint use, fire exposure, vehicle damage, and vehicle age—rather than demographic or environmental variables alone. These insights reinforce the importance of crash dynamics and occupant protection mechanisms in injury outcomes.

While the models demonstrate strong classification performance in a controlled evaluation setting, several limitations remain. Data quality issues such as missingness and encoding inconsistencies can introduce bias or noise. Additionally, the binary classification approach simplifies a complex, ordinal injury severity scale, and the current models are trained on a filtered subset of the broader CRSS dataset.

To ensure that model outputs are reliable and beneficial in practice, technical deployment must be coupled with ongoing model validation, performance monitoring, and ethical safeguards—especially when used in public safety, insurance, or law enforcement applications.