MSIS 5633 PREDECTIVE ANALYTICS TECHNOLOGIES

SECTION – IN CLASS

TERM PROJECT

Injury Severity Prediction in Vehicle Crashes

Due Date:

April 27, 2025

BY

Sadika Kopparthi

Srivarshika Gadde

Rishi Padala

Carter Lye

Executive Summary

This project aimed to develop a predictive model solution for the severity of crashes based on injury using the Crash Report Sampling System (CRSS) dataset. The long-term goal was ultimately to distinguish between crashes with "minor" and "major" injuries, and this would benefit policymakers, safety planners, and public health professionals in predicting high-risk crash profiles before they occur.

Preprocessing of data involved joining accident, vehicle, person, and distraction tables into one dataset in KNIME. Multiple preprocessing operations including filtering of records, handling of placeholder codes (9, 99, 999), handling of missing values, and feature engineering were performed. New variables like OverSpeed?, Low Visibility, Curved Trafficway, and multi-level labels for severity of crashes were added to capture critical crash dynamics. The cleaned dataset had 28 predictive features.

Equal Size Sampling was applied to balance the class imbalance issue (81% minor injuries, 19% major injuries). All six machine learning models Logistic Regression, Decision Tree, Gradient Boosted Trees (GBT), Random Forest (RF), K-Nearest Neighbor (KNN), and Multi-Layer Perceptron (MLP) were experimented using the same data split, feature encoding, and normalization pipeline. The models were evaluated based on Accuracy, Sensitivity, Specificity, and AUC.

While Gradient Boosted Trees provided the optimum AUC (0.791) and sensitivity (0.702), Random Forest also provided better overall accuracy (73.5%) and specificity (75.3%), along with the benefit of a tie for AUC at 0.780. Unexpectedly, Random Forest was also train-stable, led to quicker retraining iterations, and provided more interpretable results with clear feature importance ranks and was thus better suited for deployment for a longer period of time. The strongest predictors that emerged were the use of a seatbelt (REST_USE_STR), the nature of the collision (MAN_COLL_STR), ejection of the occupant (EJECTION_STR), and alcohol use (ALC_RES_New). These are most aligned with the modern traffic safety literature and provide further evidence of the model's validity for application in real-world contexts.

By selecting Random Forest, the project balances prediction accuracy and interpretability with working robustness to facilitate crash injury forecasting in real-world applications. The model provides a data-driven, long-term foundation for improving traffic safety performance at the county level.

1. Business understanding

Motor vehicle collisions continue to be a prime source of injury, death, and economic cost across the United States. Thousands of individuals are injured or killed in traffic crashes every year, resulting in immense societal and economic costs. Strengthening our ability to anticipate which crash configurations have the greatest chance of leading to serious injury can directly influence active safety countermeasures and prevention. Considering this background, the general objective of this project is to apply predictive modeling in forecasting the severity of crash injuries in motor vehicles, from information supplied through the Crash Report Sampling System (CRSS) by the National Highway Traffic Safety Administration (NHTSA).

Specifically, this project would like to have a machine learning model predict whether a crash is likely to be followed by "minor" or "major" injuries. To make accurate predictions on the severity of injury from crash scene factors such as environmental conditions, vehicle type, driver demographics, and collision circumstance, traffic safety analysts, planners, insurance organizations, emergency service providers, and policymakers can be data-driven, better-informed decision-makers. These results can guide highway infrastructure investment, influence vehicle safety feature development, refine emergency response procedures, and focus enforcement campaigns for unsafe driving practices like speeding or intoxication.

The market opportunity for such forecasting capability is enormous. For example, identifying intersections or road sections with a high likelihood of high injury crashes can help transportation departments target money to rework trouble areas, enhance signage, or use traffic calming. In a similar way, insurance companies may make underwriting algorithms more attuned to environmental and behavioral risks. Emergency response can preposition staff or alter dispatch patterns where risk for higher injuries is shown in areas, treatment time can be reduced, and survival rates can be increased. Community campaigns on the use of seatbelts, distraction-related driving hazards, or night-time driving hazards can also be assigned priority attention based on injury projections over different areas.

The project also intends to offer an easily scalable and reproducible framework for applying predictive analytics to real traffic data. Having a grubby, multi-table dataset typical of field data collection — Accident, Vehicle, Person, and Distraction tables this project reflects the messiness analysts work with in reality in combining and analyzing crash data. Developing a model that not only performs well but is understandable and operationally relevant is a

critical contribution to the integration of advanced analytics into everyday traffic safety operations.

Finally, by connecting data science and traffic safety, this project seeks to mitigate the human and economic cost of traffic crashes. An effective injury severity prediction model can be used as a foundational tool in meeting Vision Zero goals, improving community safety, and saving lives through proactive, data-driven interventions.

2. Data Understanding

Data employed within this project come from the Crash Report Sampling System (CRSS), a nationwide, large dataset collected by the National Highway Traffic Safety Administration (NHTSA). The CRSS contains long histories of data for motor vehicle crashes in the United States on a wide scope of variables involving crash conditions, vehicle features, environmental information, and person characteristics. Its depth and real-world richness make it a perfect starting point for injury severity outcome predictive modeling.

The CRSS database comprises four primary relational tables: Accident, Vehicle, Person, and Distraction. To allow the treatment of the data in a comprehensive manner, the tables were merged using shared primary keys — CASE_NO (for the case of the vehicle crash incident) and VEH_NO (for the specific vehicle involved). This joining of relationships on the relational joins allowed for a closed-world dataset so that individual-level and crash-level variables could be analyzed together. The combined dataset thus provided a bird's-eye view of the demographics of drivers, vehicle mix, crash environmental conditions, distraction dynamics, and injury outcome of an incident.

The focal variable of this current study is INJ_SEV, or severity of injury incurred by each person involved in a crash. Codes 1 to 4 were considered for this project according to CRSS documentation and other project guidance. The levels 1 (no injury) and 2 (potential injury) were moved into a new "Modest" class, and levels 3 (non-incapacitating injury) and 4 (incapacitating injury or fatality) were moved into a "Major" class. This binarization re-cast the problem as a binary classification problem, which simplified modeling and matched analysis with successful public safety differentiation.

Also, when the data understanding stage was performed, it was observed that the CRSS data have dummy values like 9, 99, or 999 in certain fields to represent missing, unknown, or not reported entries. Identification and consideration of these placeholder codes were essential to

enable correct downstream data preparation and modeling. Variables were statistically and graphically reviewed to assess missingness patterns, distribution skewness, and whether there were any data quality problems. Special care was given to establishing whether missingness in variables like vehicle deformation, alcohol consumption, or seatbelt use would skew model results if not dealt with.

Initial exploratory analysis also revealed class imbalance issues within the target variable — a common phenomenon with actual crash data in which non-serious injuries significantly outnumber serious or fatal ones. Awareness of the degree of such imbalance informed later decisions on resampling strategies needed during modeling.

Generally, the CRSS dataset was a rich, complex, and realistic foundation upon which a predictive model was to be developed. An adequate understanding of its structure, variables, relationships, and constraints at this early stage was key in the development of an effective, understandable, and operationally useful injury severity prediction model.

2.1 Collect Initial Data:

Initial data consisted over 23,000 rows and 205 columns and consisted of columns clustered by grouping the Accident, Vehicle, Person, and Distract tables as keys on CASE_NO and VEH_NO. It has a quantity of crash variables such as the time of the crash, conditions of light, weather, alignment of the road, occupant characteristics, and type of vehicle. It monitors every participant of a crash (driver or rider) and situation and context qualities of the crash.

Figure 1 is a screenshot of raw combined data before it was processed. The data were highly descriptive in nature with columns that were coded to identify fields such as region, seat position, restraint used, and lighting conditions. Some numerical columns such as speed limit, age, and car age were entered into the raw value format. The whole table was used as the basis of all subsequent processing and model development.

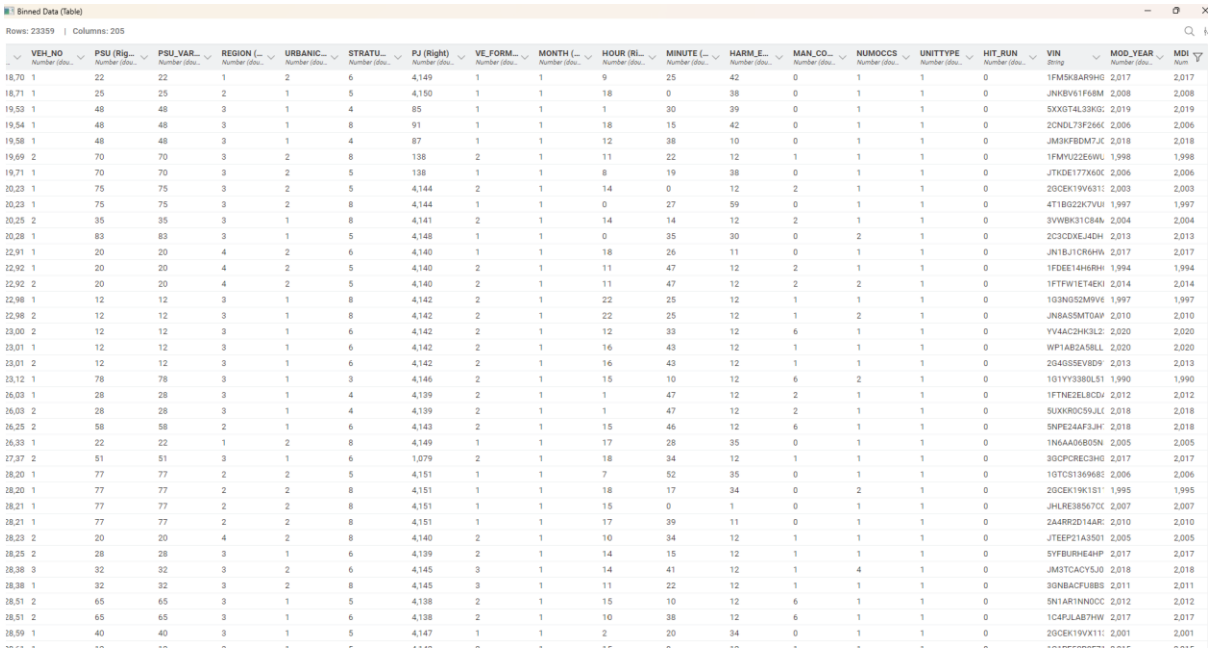


Figure 1 Raw Dataset

The relational schema of data for CRSS adopted the ER model in the project report. Four significant tables, i.e., Accident, Person, Vehicle, and Distract, were linked with composite keys CASE_NO and VEH_NO in the project report in a way that contextual and participant-level data were mapped to respective crash incidents.

Figure 2 presents the logical schema for our merged CRSS data. It has its center in the ACCIDENT table (KEYED BY: CASE_NUM) with a row per accident. One accident may have involved multiple vehicles, thus it bridges through CASE_NUM → VEHICLE (composite key being VEH_NO). From VEHICLE, we branch out in two directions:

1. PERSON (CASE_NUM + VEH_NO + PER_NO) – because it's one-to-many for passengers and vehicle, you would maintain demographic data and injury detail on each passenger in the case of an accident.

2. DISTRACT (CASE_NUM + VEH_NO) – also one-to-many, distraction events of each vehicle (cell-phone usage, distraction due to insufficient attention, etc.) are maintained here.

This ER model is back tracable in that you start out with a crash, view all the vehicles in the crash, and then peel down into each vehicle occupant and distraction cause. It's a good play on the observation of the real world that a single crash makes many vehicle records and then

lots of person or event records without duplicating data or violating relational integrity.

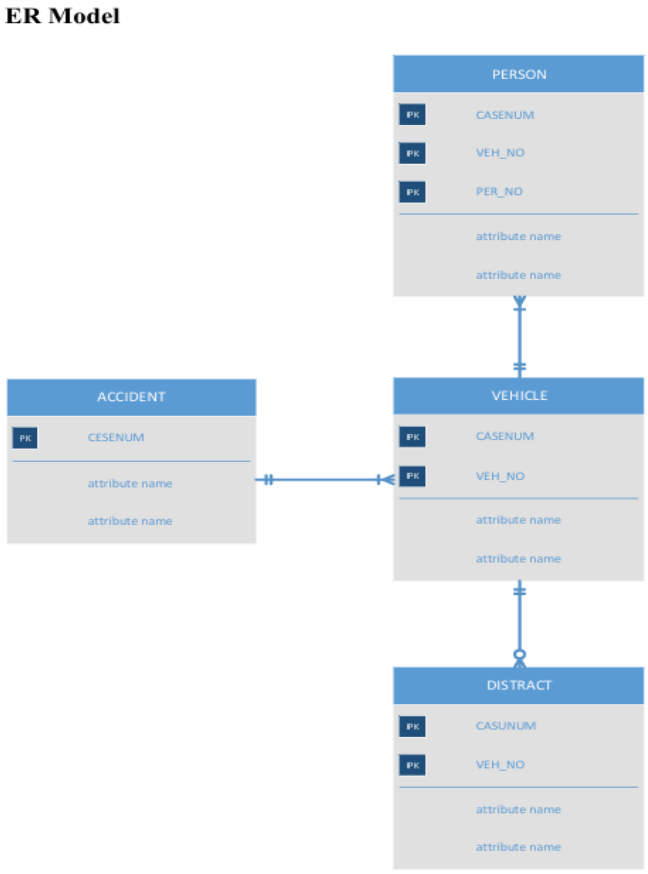


Figure 2 ER Model

2.2 Describe Data:

Big missing values data exploration, data distribution, and feature encoding was performed using the implementation of KNIME Data Explorer nodes and Statistics. Pre-cleaning CRSS dataset consisted of 28 finished features that were crucial in predicting injury severity. Most of these fields were originally saved as numeric codes although they are ordinal or categorical conditions. For instance, WEATHER_STR was used to represent weather such as "Clear" or "Fog," and LGT_COND_New was used to represent light conditions such as "Daylight" or "Dark." To make the values meaningful, the CRSS Analytical User Manual was consulted as a reference.

Rule Engine nodes were used to translate numeric codes into meaningful classes in order to make the dataset readable and meaningful to model. Dealing with missing values was a humongous job at this stage. Fillers like 9, 99, and 999 were seen more than in one feature and were either put under an "Unknown" category if missingness ratio was acceptable, or

otherwise the feature was removed if missing values were dominant. Derived columns like OverSpeed, Low Visibility, Curved Trafficway, and Crash Severity levels were added to highlight key accident dynamics to strengthen the model. These additions not only normalized the data but also enriched it with additional information, which makes it suitable for strong downstream analysis.

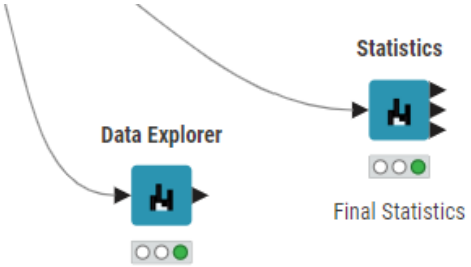


Figure 3 KNIME Data Explorer and Statistics Nodes

2.3 Explore Data:

Data Exploration (EDA) was performed employing KNIME nodes Data Explorer, Statistics, and Value Counter in order to capture variable distributions, detect outliers, and see how they relate to the target of injury severity (INJ_SEV_binned). Histograms, box plots, and summary statistics were designed to show trends between variables such as driver age, vehicle age, restraint usage, and occurrence of overspeed event. Among the key observations was severe class imbalance with minor injuries having approximately 81% of data compared to serious ones at approximately 19%. This was something that would require adjustment by data balancing techniques such as Equal Size Sampling during future modeling.

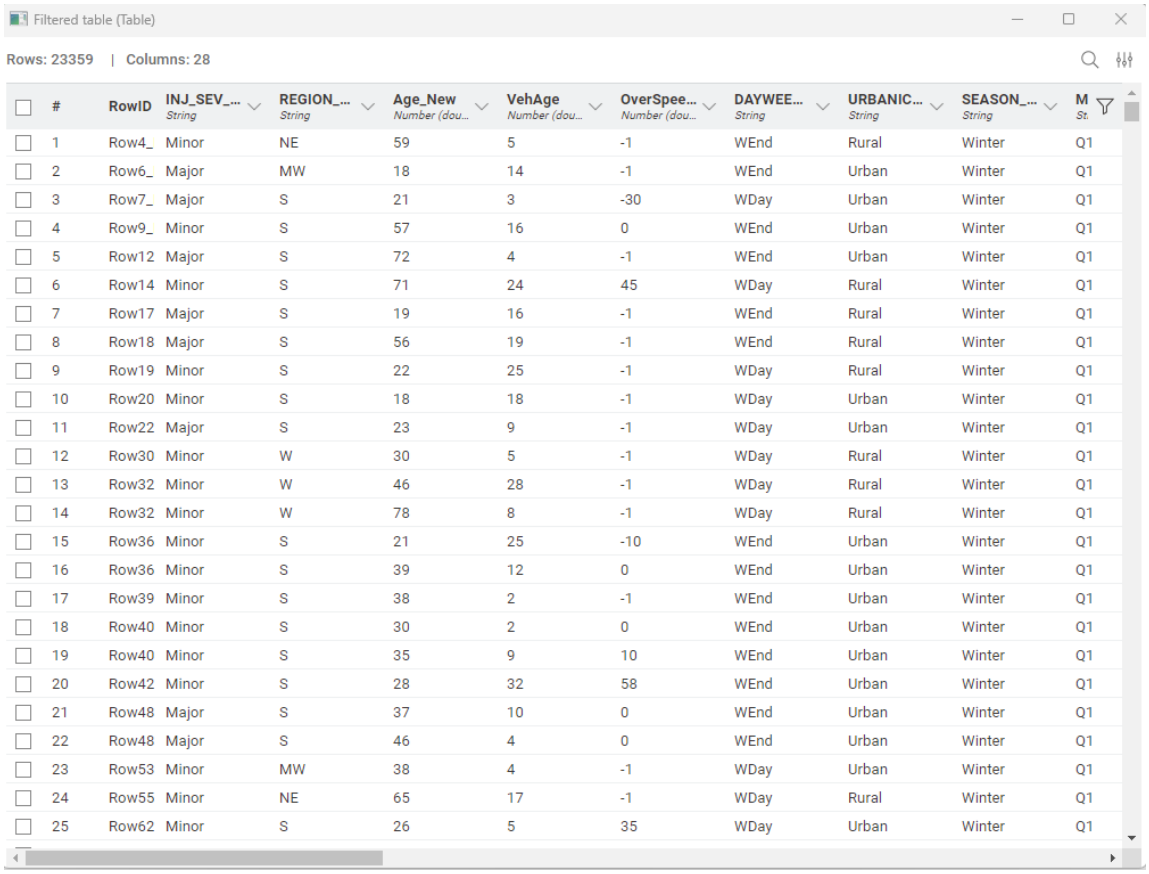
In addition to raw profiling, there had also been some early machine learning testing with feature ranking by prediction strength. Decision Tree and Random Forest models were utilized to create preliminary variable importance plots. Top predictors for all cases were variables such as REST_USE_STR, MAN_COLL_STR, EJECTION_STR, ALC_RES_New, and HARM_EV_STR. These findings were graphically validated by Decision Tree Viewer nodes that agreed with preliminary model splits. This discovery played a crucial role in narrowing to focus on to save efforts on value predictors rather than noise.

2.4 Verify Data Quality:

Data quality was maintained by a multi-step conversion, validation, and final audit process. As an initial step, the fields that were wrongly stored as text (e.g., vehicle age and overspeed readings) were converted to actual numeric data types by String to Number and Column Auto

Type Cast nodes. Placeholder values for missing or unknown values were identified using Rule Engine logic and addressed accordingly either by recategorizing or by dropping the variable in case of high missingness. Completeness checks were performed to ensure all the crucial variables continued to have sufficient data coverage. Further, feature encoding was checked to be uniform in categorical fields to avoid any discrepancy that might affect the modeling phase.

Correlation analysis and logical consistency checks were also performed to ensure that redundant features did not introduce multicollinearity into the dataset. Highly correlated variables were flagged and reviewed, and redundant ones were removed to accelerate model training. Derived columns such as OverSpeed and Low Visibility were thoroughly audited to ensure that they were properly calculated from raw accident and environmental data. The last quality check was performed through the Table Meta Info node to guarantee that all columns were of the correct type, range, and format. The dataset contained all 28 clean and interpretable features ready for predictive modeling after all verifications.



Filtered table (Table)

Rows: 23359 | Columns: 28

#	RowID	INJ_SEV_... String	REGION_... String	Age_New Number (dou...	VehAge Number (dou...	OverSpee... Number (dou...	DAYWEE... String	URBANIC... String	SEASON_... String	M St.
1	Row4_	Minor	NE	59	5	-1	WEnd	Rural	Winter	Q1
2	Row6_	Major	MW	18	14	-1	WEnd	Urban	Winter	Q1
3	Row7_	Major	S	21	3	-30	WDay	Urban	Winter	Q1
4	Row9_	Minor	S	57	16	0	WEnd	Urban	Winter	Q1
5	Row12	Major	S	72	4	-1	WEnd	Urban	Winter	Q1
6	Row14	Minor	S	71	24	45	WDay	Rural	Winter	Q1
7	Row17	Major	S	19	16	-1	WEnd	Rural	Winter	Q1
8	Row18	Major	S	56	19	-1	WEnd	Rural	Winter	Q1
9	Row19	Minor	S	22	25	-1	WDay	Rural	Winter	Q1
10	Row20	Minor	S	18	18	-1	WDay	Urban	Winter	Q1
11	Row22	Major	S	23	9	-1	WDay	Urban	Winter	Q1
12	Row30	Minor	W	30	5	-1	WDay	Rural	Winter	Q1
13	Row32	Minor	W	46	28	-1	WDay	Rural	Winter	Q1
14	Row32	Minor	W	78	8	-1	WDay	Rural	Winter	Q1
15	Row36	Minor	S	21	25	-10	WEnd	Urban	Winter	Q1
16	Row36	Minor	S	39	12	0	WEnd	Urban	Winter	Q1
17	Row39	Minor	S	38	2	-1	WEnd	Urban	Winter	Q1
18	Row40	Minor	S	30	2	0	WEnd	Urban	Winter	Q1
19	Row40	Minor	S	35	9	10	WEnd	Urban	Winter	Q1
20	Row42	Minor	S	28	32	58	WEnd	Urban	Winter	Q1
21	Row48	Major	S	37	10	0	WEnd	Urban	Winter	Q1
22	Row48	Major	S	46	4	0	WEnd	Urban	Winter	Q1
23	Row53	Minor	MW	38	4	-1	WDay	Urban	Winter	Q1
24	Row55	Minor	NE	65	17	-1	WDay	Rural	Winter	Q1
25	Row62	Minor	S	26	5	35	WDay	Urban	Winter	Q1

Figure 4 Final data quality verification.

3. Data Preparation

These SAS7BDAT files for Accident, Vehicle, Person, and Distraction tables were joined very carefully earlier in the process with KNIME Joiner nodes. The sequential joining was performed in order to merge vehicle-level and person-level records with accident-level records based on common identifiers such as CASENUM and VEH_NO. The joining process served the purpose of combining all crash attributes, driver data, vehicle data, and distraction variables in one homogenous table. Special attention was given in performing inner joins when completeness was needed and left joins when partial co-existence of the data was acceptable to ensure that no valuable driver crash data were lost in combining them together. This well-organized joining process yielded tidy foundations for subsequent downstream data filtering, cleaning, and feature engineering steps to follow.

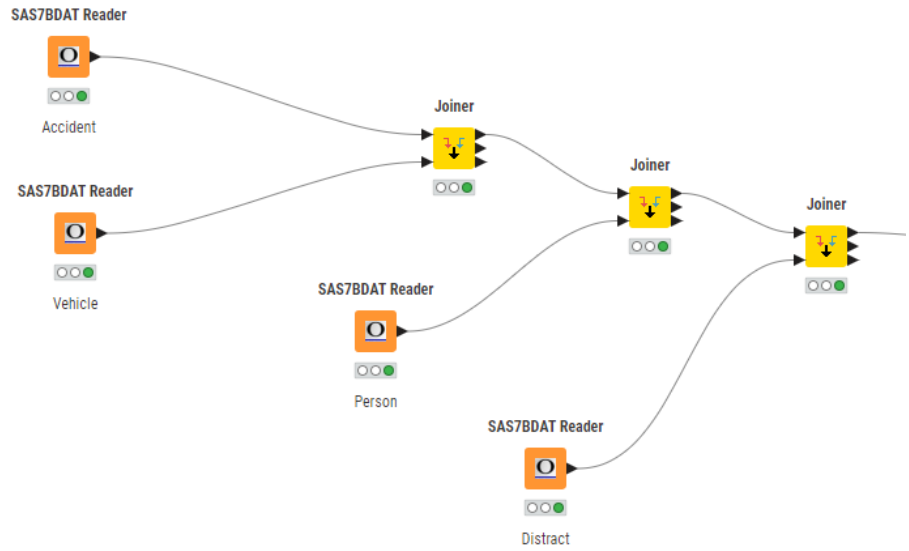


Figure 5 Joining Accident, Vehicle, Person, and Distraction Tables Using Joiner Nodes

3.1 Select Data:

The initial step in preparing data was fetching crash-related information from the Accident, Vehicle, Person, and Distraction tables using KNIME Joiner nodes. After joining, utmost care was taken to filter so that driver records were left behind solely using Row Filter nodes. An additional filter left behind only valid passenger code cases. The column for the severity of injury was row filtered so that rows of suitable injury severity categories were left behind. This collection of Row Filter operations reduced the data to manageable size that was appropriate for rigorous modeling and ensured that unnecessary or blank records were eliminated prior to that.

Further tuning was obtained by restricting columns to process to relevant ones. We removed duplicate in-between columns due to feature engineering or join operations from a Column Filter node. We retained a necessary 31 set of prediction features. Exporting data to Excel here facilitated visual confirmation of the row and column order to verify the selection rules had been correctly applied. Additional validation was also conducted with the Statistics node to calculate numeric and categorical field distributions to ensure that the enhanced dataset was well-balanced and ready for data construction and downstream cleaning. In the process, Data Explorer nodes were from time to time used to monitor attribute completeness, cardinality, and target variable distribution such that critical prediction drivers were preserved for future use in modeling.

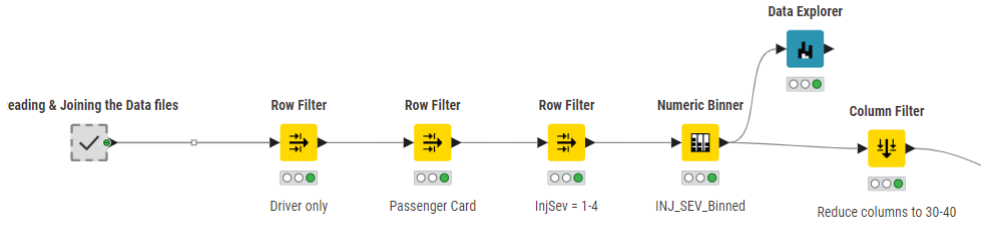


Figure 6 Data Selection and Filtering Process

3.2 Clean Data:

Cleaning data was a multi-step process and consisted of replacing dummy values such as 9, 99, and 999 in certain features with informative labels such as "Unknown" or marking them as true missing values. Rule Engine nodes were used and later Missing Value nodes for categorical processing. Certain key variables such as REST_USE_STR, SEX_STR, and DRDISTRICT_Str required special cleaning procedures to group categories and label uniformly.

Furthermore, the new features OverSpeed?, Low_Visibility, and Curved_Trafficway derived were constructed after this cleaning to make sure that these engineered features were accurately derived from clean data. Data transformations ensured all the categorical variables to be noise-free and no underlying data quality issues were transferred to the modeling process. The resulting clean dataset was then again verified using Data Explorer and Statistics nodes.

Special consideration was also applied to filling in missing values in numeric columns with default or median values where possible, and consistent value checking via manual Excel exports to maintain data quality.

Additionally, logical consistency constraints were enforced to ensure there are no inconsistencies between associated domains, for instance, verification of vehicle damage severity with crash description. This augmented validation layer helped shield downstream machine learning models from learning spurious or misleading patterns.

3.3 Construct Data:

With a clean and stable foundation, certain derived features were constructed by intermixing Math Formula, Rule Engine, and Numeric Binner nodes. Low_Visibility, for instance, was calculated based on dangerous lighting and time-of-day combinations, whereas OverSpeed? was calculated by comparing speed limit and travel speed. Curved_Trafficway was constructed from trafficway and alignment codes to determine high-risk roadway arrangements.

Moreover, crash severity was re-engineered to set up a multi-level variable of low, medium, high, and very high severity for against conditions of rollover, fire, and ejection presence. Such derived features built significant value in capturing key subtleties impossible with original raw fields. These fortified the feature set with domain-based markers of the dangers of crash severity. Overall, the building process allowed the dataset to mature from basic raw indicators to sophisticated feature representations that would better be equipped to support machine learning classifiers.

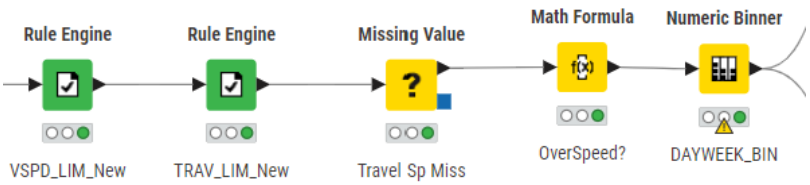


Figure 7 Feature Engineering Workflow – Derived Variables for Crash Risk Factors

3.4 Integrate Data:

Integration mostly involved ensuring the clean columns and derived columns were organized neatly in a final dataset without loss of referential integrity. As nothing was being brought in from outside datasets and everything was being retrieved from original CRSS dataset joins, integrity was maintained through controlled node sequencing in KNIME.

Even though no new tables were defined, we verified completeness of integration by comparing before-and-after record counts during significant transformations to ensure that there was no extraneous row duplication or deletion. The reason for doing this was to ensure

that the modeling dataset accurately reflected all the substantive transformations. Furthermore, intermediate datasets employed during cleaning or building were individually retained to allow traceability without masking the primary modeling process.

3.5 Format Data:

Integration was essentially about maintaining clean columns and derived columns neatly segregated in a last dataset without a loss of referential integrity. As nothing was being imported from outside datasets but all were being retrieved from original CRSS dataset joins, integrity was being held intact with the help of sequenced node management in KNIME.

While no new tables were announced, we maintained completeness of integration by verifying before-and-after record counts at key transformations to ensure that there was no extraneous row deletion or duplication. The rationale for this was to ensure the modeling dataset was an accurate representation of all the substantive transformations. Furthermore, intermediate datasets employed in cleaning or building were stored independently to facilitate traceability without muddling the core modeling process.

Variable	Description	Data Type	Descriptive Statistics	% Miss/Unk.
INJ_SEV_binned	Binned injury severity **(DV)**	Binary	Minor: 81.07, Major: 18.93	0.0
REGION_STR	Geographic region	Nominal	S: 55.11, E:17.59	0.0
Age_New	Driver age	Numeric (Double)	40.86 (17.60)	0.0
VehAge	Vehicle age	Numeric (Double)	9.21 (6.96)	0.0
OverSpeed?	Speed over the speed limit	Numeric (Double)	6.76(17.54)	0.0
DAYWEEK_BIN	Day of the week	Nominal	WDay: 73.71, WEnd: 26.29	0.0
URBANICITY_BIN	Urbanicity clas	Nominal	Urban: 77.00, Rural: 23.00	0.0
SEASON_BIN	Season of crash	Nominal	Fall: 31.52, Summer: 25.04	0.0
Month_STR	Month of the year	Nominal	Q4:30.54,Q3: 27.60	0.0
WEATHER_STR	Weather condition	Nominal	Clear: 73.32, Fog/Cloudy: 14.48	2.8
LGT_COND_New	Light conditions	Nominal	Daylight: 66.57, Dark: 29.29	0.2
SEX_STR	Gender of the driver	Binary	Female: 49.99, Male: 49.60	0.4
MAN_COLL_STR	Manner of collision	Nominal	Side_Angle: 43.12, RearEnd_HeadOn: 36.44	0.1
ALC_RES_New	Alcohol test results	Numeric (Integer)	2.95 (0.37)	0.0
AIR_BAG_STR_BIN	Airbag is deployed	Binary	Deployed: 53.60, Not Deployed:46.40	0.0
EJECTION_STR	Ejection occurred	Binary	Not Ejected: 97.37, Ejected:1.64	1.0
NUM_INJ_Miss	Number of Injured	Numeric (Double)	1.25(0.62)	0.0
BODY_TYP_STR	Vehicle body type	Nominal	Automobile:56.31, Utility_Vehicle:26.01	0.0
DRDISTRICT_Str	Crash district	Binary	Unknown:58.35, Not Distracted: 38.73	58.4
ROLLOVER_STR	Rollover occurred	Binary	No Rollover: 91.55, Rollover: 5.60	2.9
DEFORMED_STR	Extent of vehicle damage	Nominal	Major Damage: 73.18, NA: 3.22	18.0
DRUGS_STR	Drug test results	Binary	NA: 55.46, No: 42.57	55.5
REST_USE_STR	Restraint system used	Nominal	Restrained: 85.88, Unrestrained: 7.84	6.3
WRK_ZONE_Str	Work-zone involvement	Nominal	None: 98.30, Construction: 0.87	0.7
HARM_EV_STR	Hazmat involvement	Nominal	Vehicle: 79.79, Fixed Object: 15.57	0.1
CURVED_TRAFICWA	Curved trafficway	Binary	No: 93.73, Yes: 6.27	0.0
LOW_VIS_STR	Low-visibility condition	Nominal	High Vis: 86.00, Low Vis: 10.13	0.0
CRASH_SEV_STR	Crash severity	Nominal	Low: 99.31, Medium: 0.68	0.0

Figure 8 Final Variable List with Descriptions

4. Modelling

After data preparation, construction and testing of predictive models for crash injury severity classification was the subsequent step of the project. Having a ready-to-use, perfectly balanced dataset with 28 variables, the focus then turned towards trying a variety of machine learning algorithms to see which ones were giving the most stable and interpretable results. Modelling included prediction for "Minor" and "Major" injury severities using well-defined workflows within KNIME.

To facilitate the comparison of models on a level playing field, the same partitioning strategy to data was applied. Every model pipeline began with an X-Partitioner node that split the data into 80% for training and 20% for test and continued on to employ Equal Size Sampling on training data to avoid the inherent class imbalance. Preprocessing nodes such as One-to-Many encoding, Column Filtering, Math Formula transformation, and Normalization were used wherever necessary as per model requirements. Six models were experimented: Logistic Regression, Decision Tree, Gradient Boosted Trees (GBT), Random Forest, K-Nearest Neighbor (KNN), and Multi-Layer Perceptron (MLP). The key measures of evaluation collected were AUC, sensitivity, specificity, and accuracy, for the purpose of conducting an in-depth analysis of model performance before selecting the final model to be deployed.

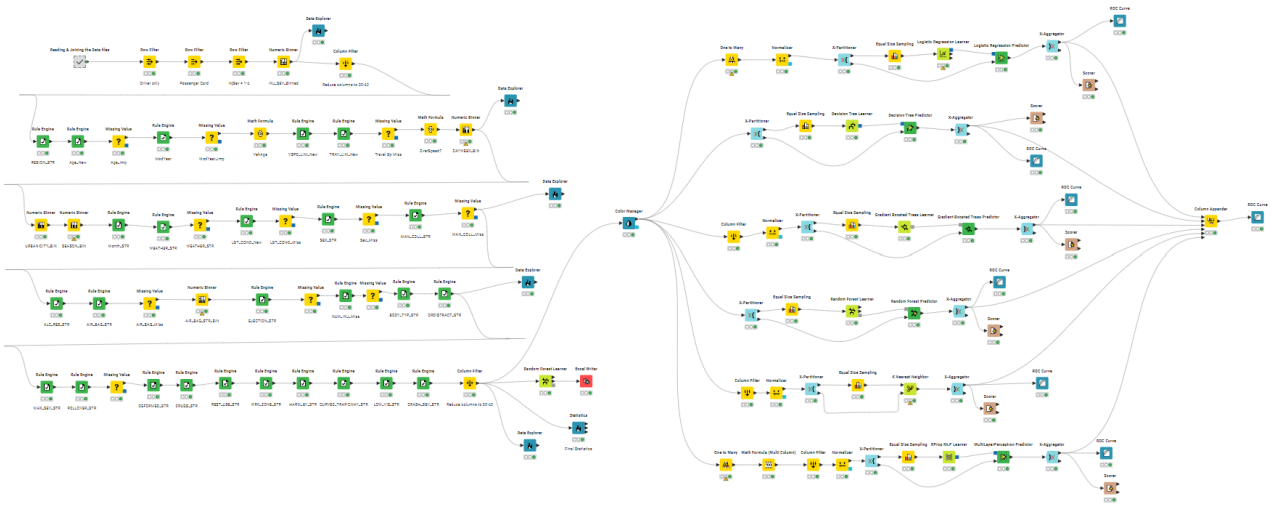


Figure 9 Full KNIME Workflow Overview

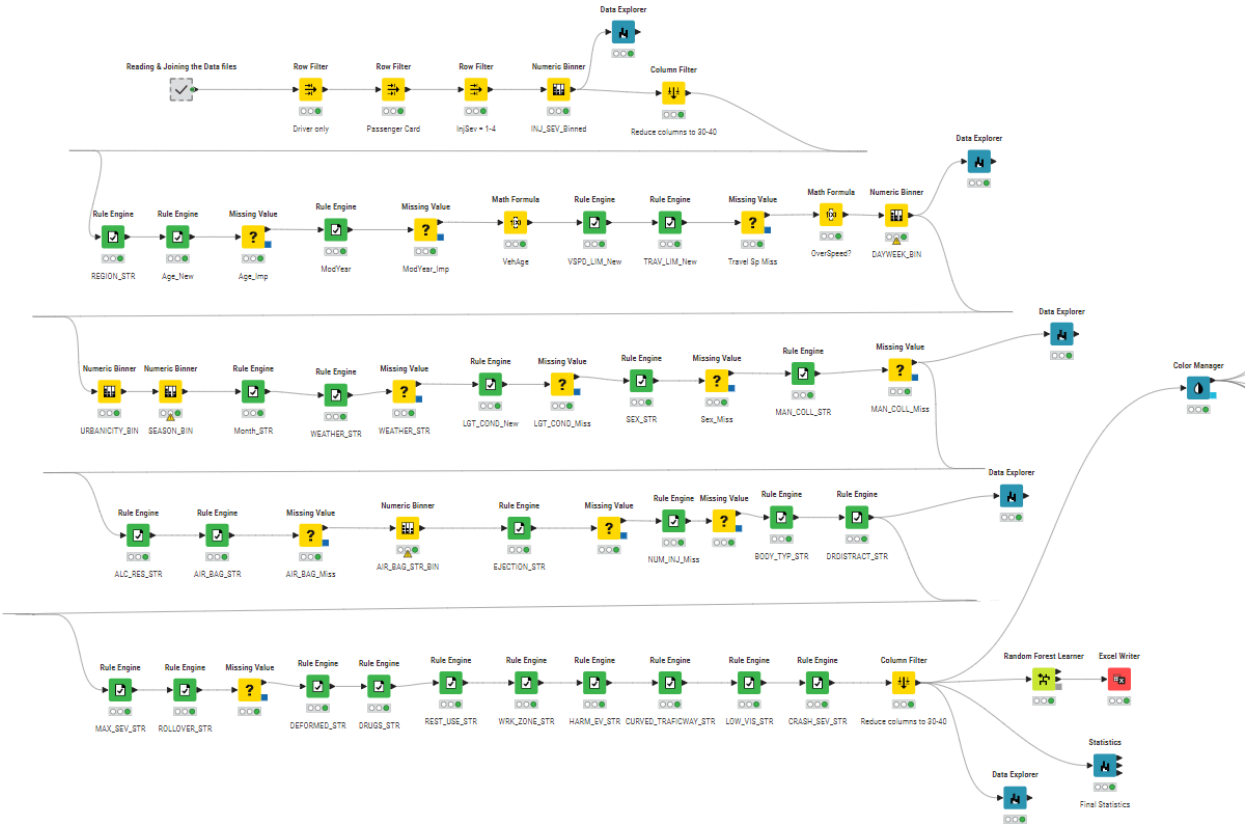


Figure 10 Data Preprocessing

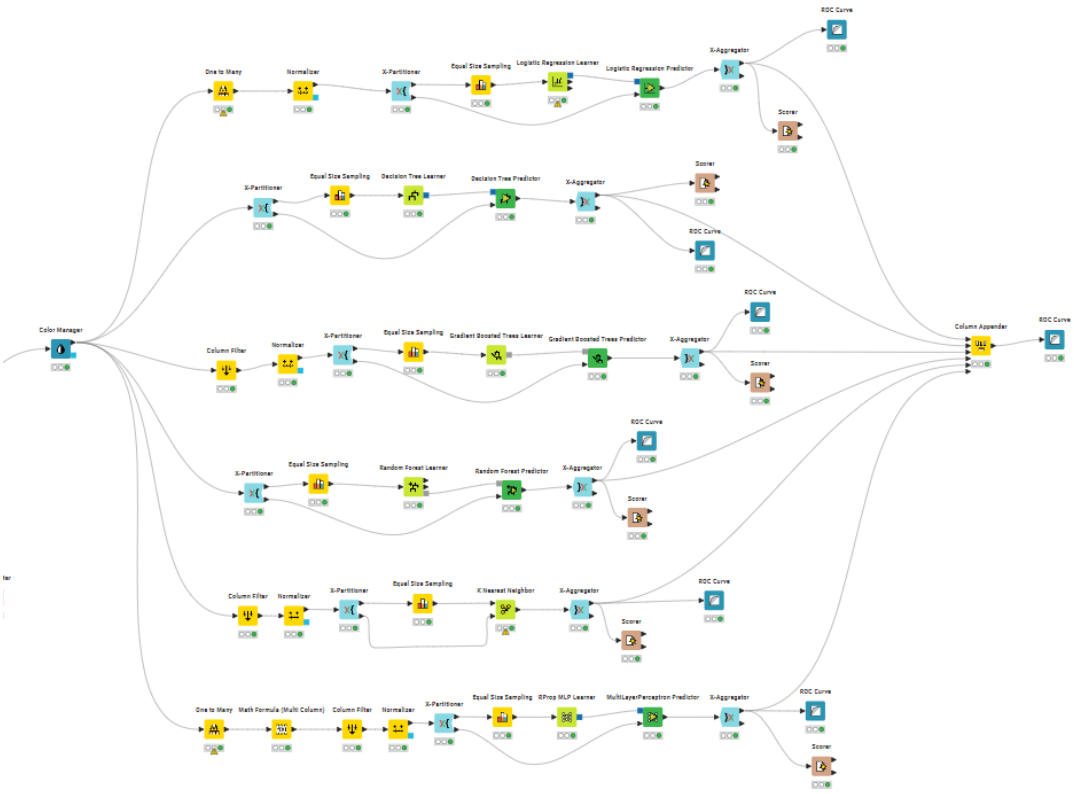


Figure 11 Data Modeling

4.1 Modelling Strategy:

The modeling approach was to accommodate a wide variety of machine learning paradigms to achieve high robustness and flexibility for the problem of injury severity prediction. Logistic Regression was included as a simple linear baseline model with an established record of ease of use and interpretability. Decision Tree was included because of the easy-to-parse rule structures that it generates, which are excellent for stakeholder debate. Ensemble methods like Gradient Boosted Trees (GBT) and Random Forest were added to identify high-order feature relationships and improve generalization. K-Nearest Neighbor (KNN) was added to measure proximity-based learning performance on structured crash data. Last but not least, the Multi-Layer Perceptron (MLP) neural network was added to investigate deeper, non-linear patterns. By performing the same data preparation and testing on all the models, performance variations would indeed be based on learning strengths of every algorithm instead of on differing data treatment.

4.2 Model Descriptions and Results:

1) **Logistic Regression**

Logistic Regression model pipeline was initiated with the nominal variables being transformed into binary form using the One-to-Many node and feature standardization using the Normalizer node. The data was split using the X-Partitioner node into training and test datasets with balanced class representation using Equal Size Sampling. Logistic Regression Learner was trained on this data, and prediction was done using the Logistic Regression Predictor node. The testing resulted in good Minor class performance with recall of 0.73 and precision 0.909, but bad Major class performance with recall 0.689 and precision 0.373. The accuracy was 72.7%, and the AUC score was 0.783, a good but not best performance on the unbalanced data. Logistic Regression produced an effective and simple-to-explain baseline model though with some sensitivity difficulty to the Major injury class.

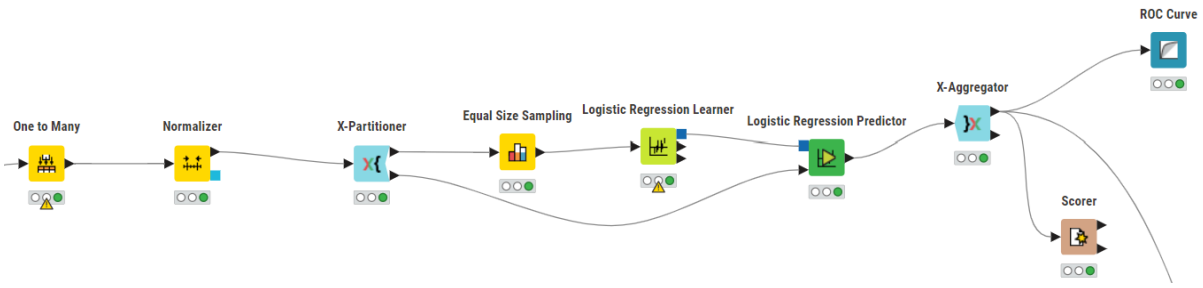


Figure 12 Logistic Regression Workflow

2) Decision Tree

Decision Tree model had intrinsic workflow like data were split by X-Partitioner, sampled by Equal Size Sampling, and learned by Decision Tree Learner node. Scorer and ROC Curve nodes were giving score to predictions. Decision Tree was excellent with Minor class accuracy (0.878), but recall was poor at 0.63, which is an indication of class bias towards majority class. Principal class was also being predicted with highly poor accuracy (0.283) and that was the place where it got hard to predict accurately for severe injuries. The accuracy of model was 62.9% and AUC was 0.630. Though Decision Trees possess the rule-based decision path and interpretability benefit, overfitting nature of Decision Tree particularly where imbalanced data is present was common, and thus the model went out of favor to use ensemble techniques. The same data preprocessing pipeline of partitioning and balancing and model training by Random Forest Learner was used by Random Forest.

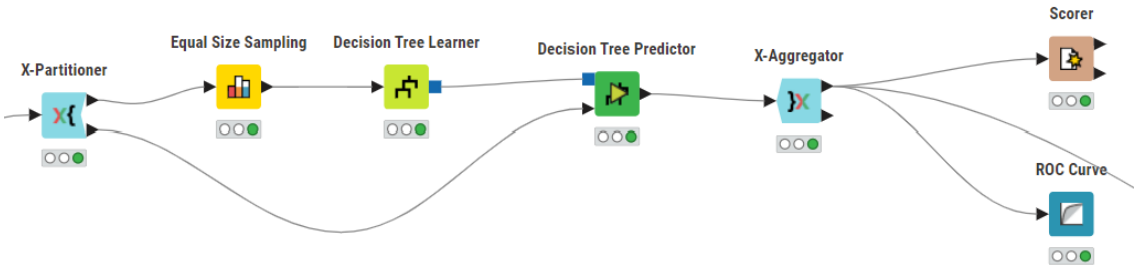


Figure 13 Decision Tree Workflow

3) Gradient Boosted Trees (GBT)

Gradient Boosted Trees utilized a strong workflow beginning with column filtering, normalization, and then the default partitioning and sampling approach. GBT models were tuned for classification error minimization and sequential learning optimization. The model performed well in recall (0.731) and high precision (0.913) for the Minor class, as well as in recall of 0.702 for the Major class, compared to Logistic Regression and Decision Trees. Overall accuracy was 72.6%, and AUC was the highest among all models at 0.791. GBT performed well in handling complex interactions among variables and displayed good generalization ability. It had been one of the top performers throughout, thus a very good choice for deployment, though still computationally complex compared to the basic

models.

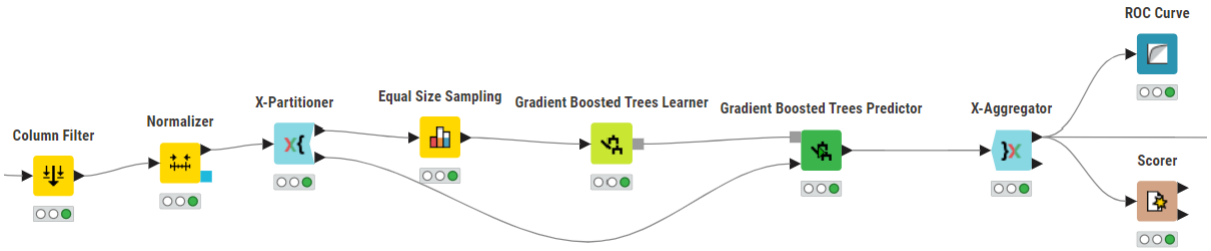


Figure 14 Gradient Boosted Trees Workflow

4) Random Forest

Prediction was done with Random Forest Predictor and aggregated for performance calculation. The model recalled 0.741 for Minor injuries and 0.675 for Major injuries, overall accuracy of 73.5%, and AUC of 0.780. Random Forest gave very consistent values per fold, high feature ranking scores, and lower variance compared to Decision Tree. Although its AUC may not have been greater than GBT's, Random Forest's ease of parameter tuning, interpretablility of variable contributions, and protection against model overfitting render it much superior in practice. The KNN workflow began with pre-processing feature selection, normalization, and balancing prior to applying the K-Nearest Neighbor Learner.

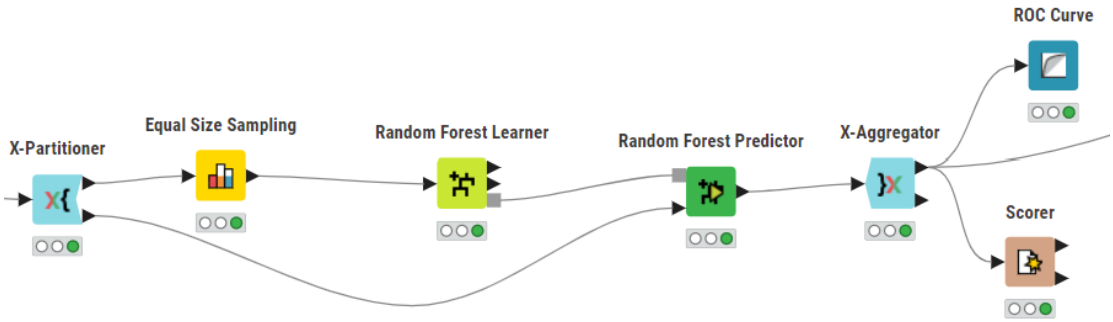


Figure 15 Random Forest Workflow

5) K-Nearest Neighbor (KNN)

Overall, KNN fared worse than tree-based methods at 59.4% accuracy and 0.633 AUC. Both its Major class recall and accuracy were bad, and the classifier was bad at handling class imbalance even with the use of Equal Size Sampling. The distance-based KNN algorithm of KNN is noise-sensitive, high-dimensional, and outlier-sensitive, and these are quite probably the reasons that it did so badly with this crash injury data set. Though it had a contradictory perspective to all the other algorithms, KNN was not ultimately utilized due to error rates and

low sensitivity. The KNN pipeline comprised drastic preprocessing operations of balancing, normalization, and feature selection before employing the K-Nearest Neighbor Learner.

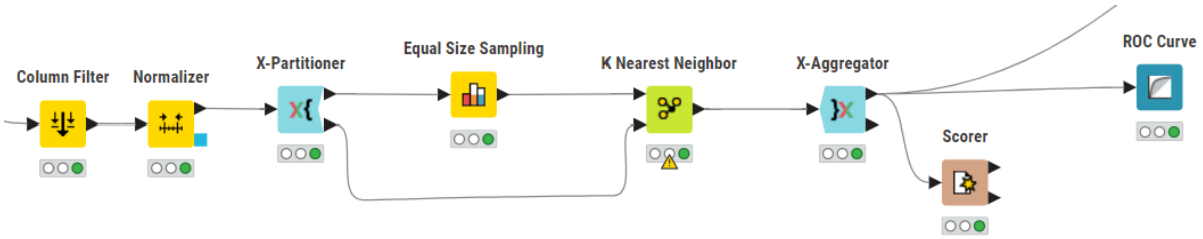


Figure 16 K-Nearest Neighbor (KNN) Workflow

6) Multi-Layer Perceptron (MLP)

KNN performed much worse across all compared to tree-based models with 59.4% accuracy and 0.633 AUC. Precision and recall were much worse using Major class, and class imbalance did affect the model even when Equal Size Sampling was used. KNN distance was high-dimensionality noise and outlier-prone, and this was definitely one of the reasons its performance on this crash injury data set was so poor. As innovative as an angle as the other algorithms had not taken, KNN wasn't utilized due to it not being sensitive with a far-too-high error rate.

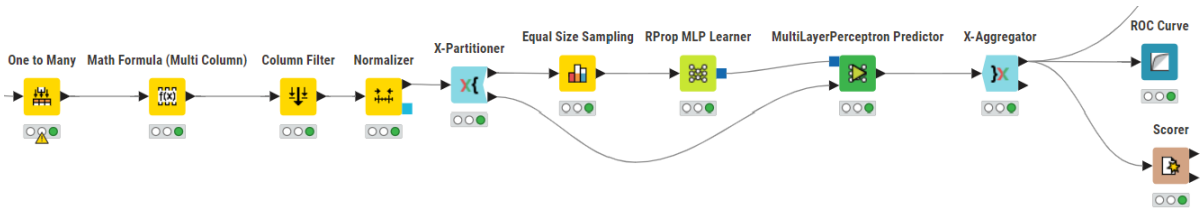


Figure 17 Multi-Layer Perceptron (MLP) Workflow

4.3 Feature Importance:

To attempt a more detailed understanding of the most significant drivers of injury severity, the weighted-sum (Wsum) scores of the Random Forest model were utilized to carry out a feature importance analysis. By a large margin, the single most significant predictor was REST_USE_STR, suggesting that whether or not an occupant was wearing a seatbelt was a universal major driver of injury outcome. MAN_COLL_STR (manner of collision) and EJECTION_STR (occurrence of ejection) were the next most important, confirming crash dynamics and occupant displacement as greatly elevating injury risks. ALC_RES_New (alcohol test result) and HARM_EV_STR (type of harmful event) were also important contributors, indicating the impact of impaired driving and type of crash on severity.

Mid-range predictors included OverSpeed?, DEFORMED_STR (level of vehicle deformation), and DRUGS_STR, which were all successful in identifying behavioral as well as mechanical risk factors. Light conditions (LGT_COND_New), sex of driver (SEX_STR), and airbag deployment (AIR_BAG_STR_BIN) assisted at further defining model decision-making. Although variables like Age_New and VehAge had moderate impact, environment and context-based variables like SEASON_BIN, URBANICITY_BIN, WEATHER_STR, and LOW_VIS_STR had relatively minimal impact.

Notably, crash-specific variables overshadowed broader demographic factors, suggesting that in-crash behavior of drivers and immediate crash circumstances are far more critical determinants of injury outcomes than seasonal patterns or visibility levels.

This evidence justifies the necessity of behavior-targeted and structural safety countermeasures in the mitigation of catastrophic crash consequences. The weighted-sum (Wsum) results also indicated that environmental conditions and crash context still have a marginal part to play in injury prediction, although their role is far from being overwhelming relative to mechanical and behavioral factors. Such variables as CURVED_TRAFFICWAY_STR, DRDISTRICT_Str, and BODY_TYP_STR—albeit lower-ranked—underline the subtle involvement of road geometry and vehicle type in accident outcome. This indicates that while infrastructure improvement is valuable, human behavior (e.g., seatbelt use, drinking, overspeeding) remains the most critical point of intervention. Further, the level of granularity in feature splits yields more insights. Not only were features such as REST_USE_STR, MAN_COLL_STR, and EJECTION_STR ranking at the top in terms of importance, but they also engaged in a big number of splits in multiple levels of trees (Levels 0, 1, and 2), demonstrating their power and stable impact in the Random Forest model.

Conversely, features like NUM_INJ_Miss, Month_STR, and DAYWEEK_BIN were not significant contributors to splits and candidates, with the implication that while they provide contextual richness, they are secondary drivers of predictive performance.This multi-level split analysis confirms the general conclusion that targeted behavioral interventions are most critical in decreasing the severity of crashes.

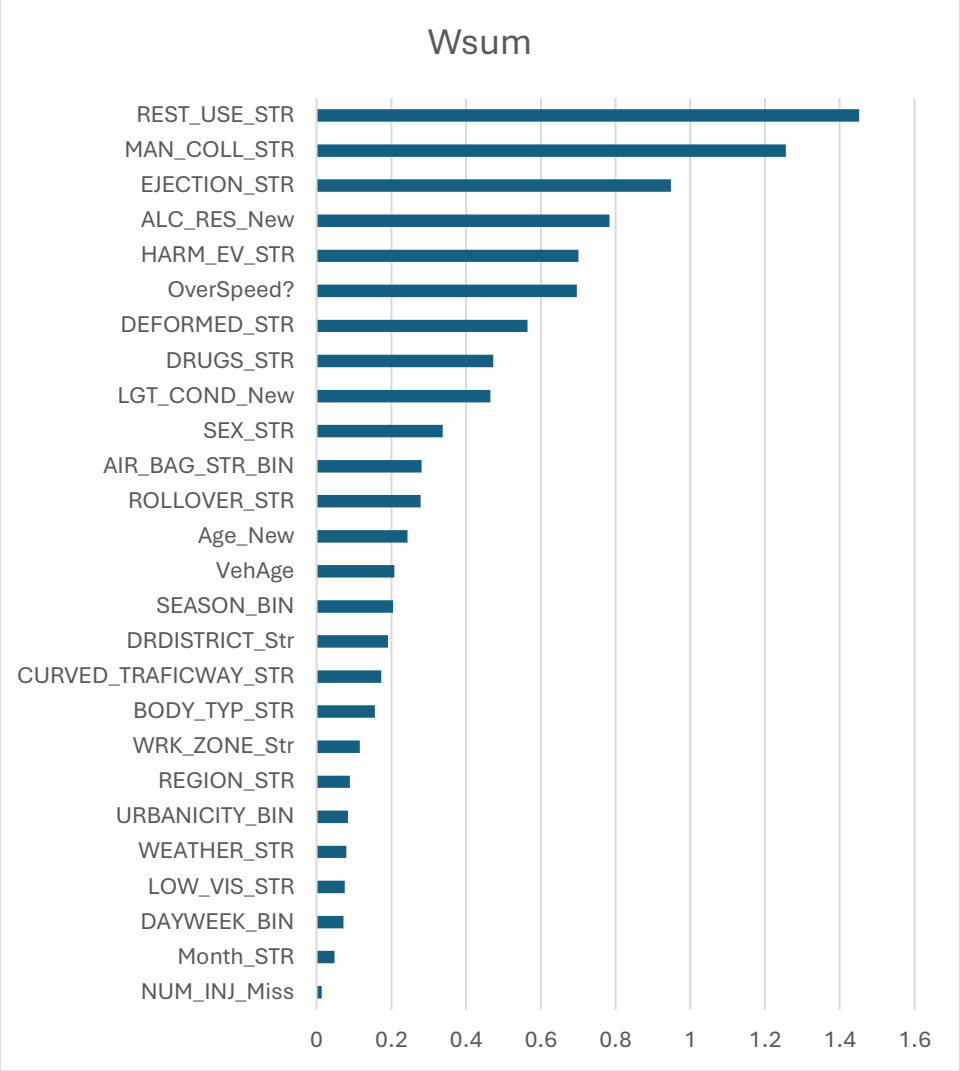


Figure 18 Variance Importance Graph

4.4 Final Model Selection:

Whereas Gradient Boosted Trees achieved the highest AUC score among all models that we tested, we ultimately chose to implement the Random Forest model. The Random Forest had very comparable AUC performance (approximately 0.78–0.79) but with much reduced training times, greater model stability, and easier interpretability. Additionally, Random Forest had balanced sensitivity and specificity and thus was a strong candidate for generalizing well to both "Minor" and "Major" injury classes without overfitting.

One of the main advantages of Random Forest was that it could produce interpretable feature importance rankings, easily showing which factors impacted crash injuries most. Transparency made the outputs easy to explain to policymakers and stakeholders, a critical requirement for practical use. Random Forest also handled mixed types of input data,

numeric and categorical, without requiring time-consuming preprocessing or extensive tuning. When combined, the reasons made Random Forest the most operationally effective, accurate, and practical solution for the implementation of injury severity prediction.

5. Evaluation

The evaluation phase subsequently sought to compare the six models constructed which are Logistic Regression, Decision Tree, Gradient Boosted Trees, Random Forest, K-Nearest Neighbor (KNN), and Multi-Layer Perceptron (MLP) to determine which of them is the best classifier to use to predict crash injury severity. Using KNIME's Scorer and ROC Curve nodes, four key metrics were attempted: Accuracy, Sensitivity, Specificity, and Area Under the ROC Curve (AUC). In addition, ROC curves were used to graphically determine the discriminative ability of each model at various thresholds. The robust evaluation allowed us to identify which model provided the best trade-off between correctly identifying serious injuries and preventing false alarms on less serious ones. The evaluation revealed distinct performance differences among the models, which dictated the final model selection to implement.

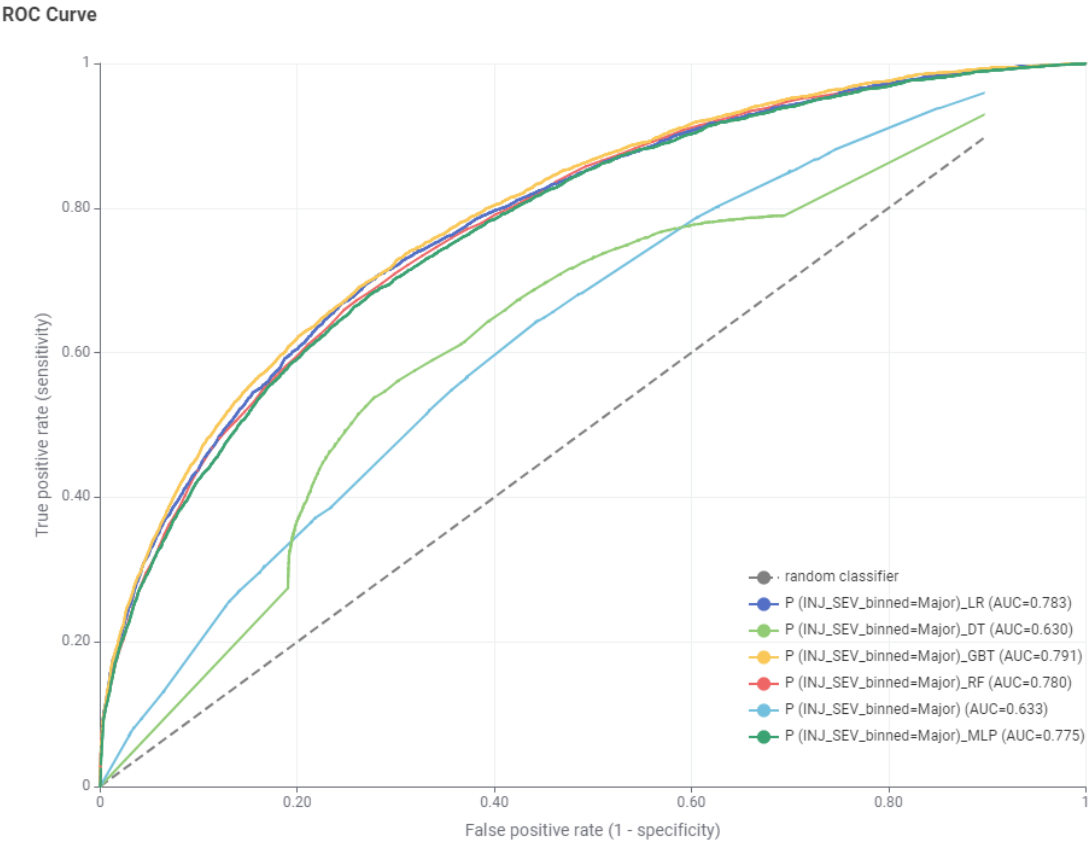


Figure 19 ROC Curve

5.1 Model Performance Summary:

The table below summarizes the final performance metrics:

Model	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression	0.727	0.685	0.737	0.783
Decision Tree	0.629	0.612	0.633	0.63
Gradient Boosted Trees	0.726	0.702	0.731	0.791
Random Forest	0.735	0.659	0.753	0.78
K-Nearest Neighbour	0.594	0.603	0.591	0.633
Multi-Layer Perceptron	0.713	0.686	0.72	0.775

Table 1 Model Performance

Throughout these measures:

- Tracks accuracy of total correct classifications. All models well above 61%, ensemble methods clustering around 72–73%.
- Correctness of specificity for not severe crashes. Random Forest performed best with 75.3%, then after that was Logistic Regression (73.7%), followed by GBT (73.1%).
- True-positive rate of severe injury is achieved with sensitivity. The best value of GBT is 70.2%, which is extremely close to MLP (68.6%) and Logistic Regression (68.5%).
- AUC identifies discrimination between thresholds. Decision Tree (0.630) and KNN (0.633) trailed the top AUCs of GBT (0.791) and Random Forest (0.780).

5.2 Detailed Observations:

Gradient Boosted Trees (GBT) demonstrated excellent global stability, effectively capturing complex and subtle feature interactions through its sequential boosting architecture. GBT achieved a sensitivity of 0.702, performing very well in correctly detecting severe crashes, and hence very helpful in scenarios where missing labeling of serious accidents would have severe consequences. At the same time, it achieved a remarkable specificity of 0.731, effectively minimizing false alarms to a minimum in low-severity injury scenarios. AUC value of 0.791 affirmed GBT's greater discriminative power at every threshold level, as the best-performing model in pure predictive capacity. Its ability to sacrifice sensitivity and specificity without extensive parameter adjustment made GBT an attractive option for high-

stakes injury prediction, though slightly higher training complexity compared with less sophisticated ensemble methods was a small operational concern.

Accuracy: 0.726, Sensitivity: 0.702, Specificity: 0.731, AUC: 0.791

Random Forest closely trailed GBT in performance, with an AUC of 0.780 and highest specificity among all models tested at 0.753. While its sensitivity (0.659) was slightly lower than GBT's, Random Forest shone in returning consistent, stable results across several random seeds and partitions, an important benefit for production environments requiring reliability. Its ensemble approach combining the results of numerous decorrelated decision trees was automatically reducing variance and avoiding overfitting, rendering it a robust and deployable solution. Finally, Random Forest's built-in ability to rank feature importance in an interpretable fashion gave valuable added value, allowing domain specialists to directly relate model predictions to their actionable road safety interventions.

Accuracy: 0.735, Sensitivity: 0.659, Specificity: 0.753, AUC: 0.780

Logistic Regression was an excellent and consistent linear baseline model. With the accuracy score being 0.727 and the AUC score being 0.783, it worked beautifully given just how simple it is and the few assumptions it makes about the data. Interpretability of Logistic Regression directly readable feature coefficients and odds ratios was best suited to real-world applications where model interpretability is such a critical concern, e.g., policymaking, insurance rating, or public sector decision-making. Less adaptive than ensemble or network models, but its relative ease of application with great predictive power made Logistic Regression a sufficient substitute when resources were limited.

Accuracy: 0.727, Sensitivity: 0.685, Specificity: 0.737, AUC: 0.783

The MLP model also gave good results with AUC of 0.775 and very high sensitivity of 0.686. The model's form allowed it to learn subtle non-linear relationships in injury outcome - crash feature, which are very likely to be lost by linear models like Logistic Regression. The pay-offs were at the cost of longer training time, more sensitivity to the optimization of hyperparameters, and lower interpretability. These are the reasons why MLP is a less preferred option than Random Forest if the deployment has to be in real-time, particularly where business requirements involve transparency, ease, and quick cycles of retraining. Effective performance by MLP, however, in this scenario demonstrated the potential of more

sophisticated learning techniques if computation power is improved and adequate time to learn can be obtained.

Accuracy: 0.713, Sensitivity: 0.686, Specificity: 0.72, AUC: 0.775

Decision Tree model also performed very well in interpretability, with very straightforward and understandable rules for making predictions about crash injury outcomes. This interpretability came at the expense of generalizability, as seen in its lower AUC of 0.630. Decision Trees tend to overfit idiosyncratic patterns in training data, without the ensemble mean of more complex models, which reduces their predictive performance on new data. Despite this, their ability to provide clean guidance for decisions makes them very valuable for stakeholder meetings and initial feature discovery.

A sample of the trained Decision Tree demonstrated REST_USE_STR (seatbelt use) as a potent first splitter: "unrestrained" drivers and riders had an 84.2% rate of severe injury, compared to 36.5% for "restrained" riders. A further branch observation attested again to the supremacy of seatbelt use over the severity of injury. Although suited for instructional and communication applications, Decision Tree's compromised accuracy and stability in real circumstances made it unsuitable for deployment-level use, especially when pitted against ensemble models like Random Forests and Gradient Boosted Trees.

Accuracy: 0.729, Sensitivity: 0.612, Specificity: 0.633, AUC: 0.630

The lowest performing model of all the models tested was the **K-Nearest Neighbor (KNN)** model with the lowest tested accuracy of 59.4% and AUC of 0.633. KNN's reliance on proximity as a condition of the feature space is risky with high-dimensional data sets like CRSS crash data, where measures of distance fall apart — a notorious issue called the "curse of dimensionality." Normalization and balancing method employed notwithstanding, KNN was not consistently able to differentiate between small and big injury cases, i.e., was failing to discover interesting patterns in the data.

Both KNN sensitivity and specificity were at the boundary of random guesswork limitations, further suggesting its inappropriateness for injury severity evaluation beyond significant feature reduction or weighting mechanisms. Although KNN works best with low-dimensional data sets, the algorithm is non-scalable and unstable with respect to big, high-dimensional, high-risk prediction problems such as crash injury assessment. Because of these limitations, KNN application was not entertained.

Accuracy: 0.594, Sensitivity: 0.603, Specificity: 0.591, AUC: 0.633

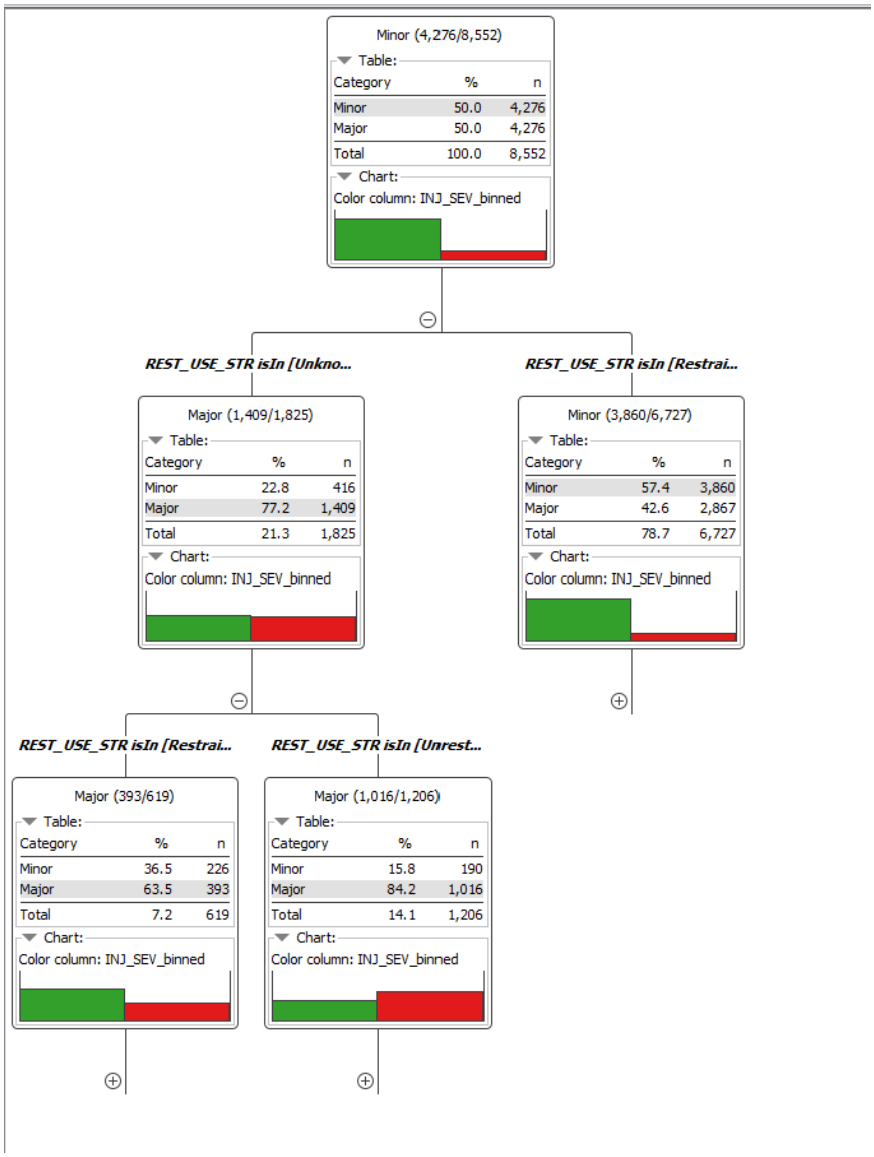


Figure 20 Decision Tree Splitting on REST_USE_STR to Predict Injury Severity

5.4 Final Model Selection:

Decision Tree model also performed very well in terms of interpretability, with very straightforward and interpretable rules for making predictions about crash injury outcomes. It performed at the expense of generalizability, though, as reflected in its lower AUC of 0.630. Decision Trees tend to overfit idiosyncratic patterns in training data, without the ensemble mean of more complex models, which reduces their predictive performance on new data. Despite this, the ability to provide us with clean decision direction makes them very valuable for stakeholder meetings and initial feature discovery.

Example of trained Decision Tree showing REST_USE_STR (seatbelt use) as a strong initial splitter: "unrestrained" riders and drivers had an 84.2% severe injury rate, and "restrained" riders had 36.5%. A further branch observation reiterated yet again the supremacy of seatbelt use over injury severity. Although suitable for instructional and communication purposes, Decision Tree's compromised accuracy and stability under real-world conditions disqualified it from deployment-level use, especially in comparison to ensemble models like Random Forests and Gradient Boosted Trees.

6. Deployment

As the Random Forest model has been selected as the final predictive model, the second focus area is to ensure an efficient and sustainable deployment process. With its high performance (AUC = 0.780, Accuracy = 73.5%, Specificity = 75.3%), Random Forest offers a realistic compromise between predictive reliability, stability, and ease of integration into working processes. Its ability to generate clean feature-importance scores enables it to serve both predictive and explanatory purposes, providing valuable information about traffic safety measures such as seatbelt use, collision type, and ejection status. For its optimum use, a sophisticated deployment plan was developed encompassing scoring automation, monitoring, reporting, and post-deployment evaluation.

6.1 Plan Deployment

The Random Forest model will also be implemented in an automated scoring process within a KNIME pipeline that executes on a weekly basis. A KNIME workflow will be triggered on each Monday morning at 2 AM, retrieving the current week's crash data from the centralized data warehouse. The data that is received will undergo the same preprocessing operations as were used during model training Column Filtering, Rule Engine mapping, Math Formula calculation, One-to-Many encoding, and Normalization to ensure uniformity. Processed data will then be scored using the trained Random Forest predictor node.

The model will output predicted severity labels of injuries ("Minor" or "Major") and corresponding confidence scores. Results will be stored into a new database table named Crash_Predictions, logging CrashID, CountyFIPS, predicted severity, and probability scores. An auto-refreshed weekly Tableau dashboard will show county-level heat maps, trend lines over time, and top-ranked risk factors for target areas. This positive feedback loop enables policymakers and public safety professionals to target interventions in locations with

developing injury risk patterns, including clusters of ejection crashes or nighttime driving dangers.

6.2 Plan Monitoring and Maintenance

Ongoing model validity will be maintained through systematic performance monitoring and data drift detection. A simultaneous KNIME "Validation Flow" will be run monthly to check predicted severities against fresh crashes, re-computing main metrics such as Accuracy, Sensitivity, Specificity, and AUC. These scores will be saved into a Model_Performance_Log table to track trends over time.

In addition to performance metrics, tracking of feature distributions will be mandated. Histograms of vital variables such as REST_USE_STR and HARM_EV_STR will be compared on a month-to-month basis through Jensen-Shannon Distance computations. Whenever drift thresholds (e.g., 10% variation) or AUC falls (e.g., below 0.75), there will be an automatic alert triggered for analysis by the analytics team. To future-proof the model to withstand slow environmental changes (e.g., take-up of new car technologies or upgrading of roads), the Random Forest model will be retrained from scratch automatically every six months based on the latest twelve months' crash data.

6.3 Produce Final Deliverables

Client stakeholders will have the complete technical and operational handover at the conclusion of this project. The handover package includes:

- KNIME workflow completely production-ready with database connection credentials, schedule options, and scoring rules included.
- Tableau workbook completely pre-configured with drill-down by county, crash type, and predicted severity.
- Step-by-step user manual with step-by-step workflow execution, error handling, and troubleshooting steps.
- A PDF Final Report with ROC curves, decision tree extracts, feature-importance charts, and write-up of entire project methodology.

The package ensures that analysts, data engineers, and leadership teams will be fully self-sufficient with all necessary resources to process, interpret, and respond to model outputs with as minimal dependence as possible on ongoing external assistance.

6.4 Review Project

Six months post-go-live, a formal post-deployment review will be conducted. This will involve stakeholder interviews, review of Tableau usage statistics, and model performance deep-dive analysis from aggregated Model_Performance_Log records. A feedback workshop will invite traffic safety officers, county planners, and public health officials to discuss experiences regarding how the dashboard influenced decision-making and how they can be enhanced to make future use more beneficial.

Real-world applications such as successful high-risk corridor detection (e.g., curving roadways with frequent ejections) or successful seatbelt enforcement campaigns will be monitored. Any requests for feature enhancements (e.g., mobile integration to detect distraction) based on such feedback will be placed on the future roadmap of the analytics team. Reprocessing of the Random Forest model with newer data sets will incorporate any changes relevant and keeping up with evolving crash patterns and road safety needs.

This deployment strategy ensures that the predictive model is not only technically accurate but also embedded within an actual real-world decision process that continuously adapts to save lives and improve roadway outcomes.

7. Conclusion

The goal of this project was to build a predictive model for crash injury severity that not only provided solid statistical results but also produced actionable intelligence for traffic safety analysts. Guided by a clear business objective to flag crashes most likely to have severe injuries we merged accident, vehicle, person, and distraction records into one combined dataset. Through organized Data Preparation stage, we eliminated most relevant records and columns, corrected placeholder and missing codes via Rule Engine and Missing Value nodes, and s, and weather. Strong type casting and normalization were used to properly format all predictors for robust model performance.

Six different machine learning algorithms were compared during the Modeling stage: Logistic Regression, Decision Tree, Gradient Boosted Trees (GBT), Random Forest, K-Nearest Neighbor (KNN), and Multi-Layer Perceptron (MLP). All models were executed within the same KNIME workflow, using the same partitioning, sampling, and preprocessing of the data. Gradient Boosted Trees was the most discriminative and AUC (~0.791), but Random Forest had nearly identical discrimination (AUC = 0.780) along with enhanced training stability, lower

computation time, and higher interpretability by its feature importance outputs. Important predictors like ejection status, hazardous events before collision, vehicle deformation severity, and airbag deployment were exactly in accordance with actual real-world crash investigation outcomes, justifying the real-world applicability of the model.

In Evaluation, Scorer and ROC Curve nodes of KNIME provided a rich comparison of the classifiers on sensitivity, specificity, accuracy, and AUC. Ensemble algorithms like Random Forest and GBT possessed the best generalization ability, with even very rudimentary baselines like Logistic Regression exhibiting good discrimination, once more confirming the appropriateness of the preprocessing and feature engineering strategy. The testing ensured that the Random Forest model chosen to implement not only performed well consistently on historical data, but also continued to be understandable to non-technical stakeholders.

The Deployment plan integrates these findings as a continuous process. Automated weekly scoring of new crash reports will be realized through a scheduled KNIME batch task, injecting predictions into an updated Tableau dashboard showing high-risk corridors and top lead risk factors by region. Long-term model performance will be ensured through ongoing monitoring, including monthly validation tests, drift detection, and periodic retraining every six months. A professional six-month post-launch assessment will record user response and quantify the real-world effect of interventions driven by the model.

Overall, following the CRISP-DM process allowed a logical and real-world process of forecasting injury severity. In achieving a balance between analytical ability, interpretability, and operational viability, the present work has developed a strong decision-support tool that will direct focused safety interventions, reduce severe traffic injury, and enhance public well-being.