Project Report

Predicting Injury Severity in Road Accidents: A Real-Time Classification Approach

David Cebulla (1922129)
Gabriel Himmelein (1649181)
Lukas Ott (1842341)
Artur Loreit (2268917)
Aaron Niemesch (1836924)

November 23, 2025

Submitted to
Data and Web Science Group
Dr. Sven Hertling
University of Mannheim

1 Application Area and Goals (0.5 pages)

Modern intelligent and connected vehicle systems, such as the mandatory European eCall service, are designed to automatically transmit crucial accident data to emergency centers when a crash occurs. These transmissions typically include location, timestamp, and the number of passengers but lack information on the severity of injuries: a critical shortcoming for emergency services, as this information is vital for prioritizing rescue efforts and optimizing medical response times.

To address this gap, we focus on leveraging historical accident data provided by the French National Interministerial Observatory for Road Safety (ONISR). This organization maintains the official "Bulletins d'Analyse des Accidents Corporels de la Circulation" (BAAC), a national database that records all injury accidents on public roads in France. The dataset captures detailed multi-table information covering the circumstances of each accident (Caractéristiques), the location and infrastructure (Lieux), the vehicles involved (Véhicules), and the individual users (Usagers). These data entries include rich contextual variables such as road category, lighting and weather conditions, vehicle type, and the user's role and behavior at the time of the accident.

Each accident record specifies the injury outcome (grav) for every participant, coded as uninjured, lightly injured, hospitalized, or killed. Although the ONISR database provides this information retrospectively (sometimes up to 30 days after the event) it represents a uniquely valuable source for supervised learning aimed at approximating these outcomes in real time.

The goal of this project is to develop a machine learning classifier capable of predicting injury severity immediately after an accident, based solely on the information that would realistically be available to emergency responders or vehicle telematics systems at the moment of impact. Such a model can significantly enhance first-responder coordination by providing an automated injury risk assessment, enabling faster triage and more efficient resource allocation.

2 Structure and Size of the Data Set

Our analysis uses the "Annual Road Traffic Injury Databases" from 2019–2023, provided by ONISR. The data is supplied in four files per year (characteristics, locations, users, and vehicles) and includes features such as road conditions, weather, and user information. Recognizing the one-to-many relationship where each accident involves multiple participants, we defined our unit of analysis at the **individual level**. The `users` table serves as the base for our dataset.

Directly after merging the accident characteristics, locations, users, and vehicle tables, and prior to any further preprocessing steps, the consolidated dataset comprised a total of 619,817 records and 71 columns. Processing the yearly data reveals an annual volume ranging between approximately 105,000 and 133,000 records per year. The final consolidated dataset was then partitioned into a training and validation set of **493,214 samples** (covering 2019–2022) and a test set of **125,505 samples** (covering 2023).

The features we have selected are listed below. They are categorized into Original Features (retained from raw data) and Engineered Features (calculated to model complex relationships). The final dataset consists of **57 features**, comprising **23 original columns** retained from the raw data and **34 engineered features** calculated to model complex relationships.

Time context
{`time_of_day, hour_sin/cos, day_of_week_sin/cos, month_sin/cos, day_of_year_sin/cos`}

Captures seasonality and daily patterns using cyclical sine/cosine transformations for time units. Also includes `time_of_day`, which groups hours into broader categories (e.g., Night, Rush Hour).

Geospatial location
{`latitude, longitude`}

GPS coordinates (WGS84) utilized for spatial analysis and mapping.

Environment & Roadway
{`lighting_ordinal, weather_ordinal, location, infrastructure, accident_situation, horizontal_alignment, reserved_lane_present, speed_limit, road_complexity_index, surface_quality_indicator`}

Combines physical site attributes (urban/rural status, alignment, infrastructure type) with risk-based ordinals for lighting and weather. Engineered indices quantify road complexity (0–10 scale) and surface quality.

Crash dynamics & Maneuvers
{`type_of_collision, initial_point_of_impact, fixed_obstacle_struck, mobile_obstacle_struck, main_maneuver_before_accident, impact_score, impact_delta`}

Describes the collision mechanics, including maneuvers, impact points, and obstacles. Derived metrics like `impact_score` and `impact_delta` quantify the relative risk based on vehicle size differences.

Vehicle attributes & Involvement
{`motor_type, vehicle_category_simplified, vehicle_category_involved_[type]`}

Specifies the primary vehicle's type and motorization. Includes binary flags indicating if specific other vehicle types (e.g., heavy trucks, bicycles, buses) were involved in the accident.

Personal attributes
{`role, sex, age, age_group, position, pedestrian_location, pedestrian_action`}

Covers demographic data (age, sex), the user's role (driver, passenger, pedestrian), seating position, and specific actions or locations for pedestrians at the time of the accident.

Safety equipment usage
{`used_belt, used_helmet, used_child_restraint, used_airbag`}

Binary variables indicating the use of protective gear (seatbelts, helmets) or the deployment of airbags.

Accident Persona Clustering
{cluster}

A categorical feature derived from unsupervised K-Prototypes clustering. It groups accidents into distinct "personas" (e.g., 0, 1, 2) based on a combination of numerical and categorical attributes to capture complex, non-linear patterns in the data.

Target / Outcome
{injury_target}

The engineered ordinal target variable classifying injury severity into three levels: 0 (Uninjured), 1 (Lightly Injured), and 2 (Hospitalized or Dead).

3 Preprocessing

Given the nature of the "Annual Road Traffic Injury Database" as a raw database output provided by ONISR in multiple separate tables with complex many-to-one relationships, extensive preprocessing was required. The pipeline is designed to transform the disjointed raw data into a singular, model-ready tabular representation.

3.1 Data Standardization and Key Generation

The initial step involved normalizing column names to English equivalents. A critical challenge was the inconsistency in user identification across years. Data from 2022 onwards included a distinct `id_user`, whereas data from 2019 and 2020 did not. To ensure a consistent unit of analysis across all years, we implemented a synthetic key generation strategy. For older data, we generated a unique identifier by combining the accident ID with a cumulative count of users within that accident (e.g., `2019000001_U1`). This ensured that every individual involved in an accident could be uniquely tracked and merged with their respective vehicle and accident characteristics.

3.2 Advanced Data Merging strategies

A user-centric view was adopted for merging, treating each participant as an independent instance. While accident characteristics (weather, time) could be joined directly, the relationship between users, their vehicles, and opposing vehicles required complex logic.

Vehicle Antagonist Resolution

A significant predictor of injury severity is the disparity between the user's vehicle and the "opposing" entity (the antagonist). Since a simple join cannot determine which of the multiple vehicles in an accident caused the injury, we engineered a selection algorithm. We assigned an `impact_score` to vehicle categories based on mass and risk (e.g., HGV Truck = 6, Bicycle = 2). For multi-vehicle accidents, the pipeline identifies the "antagonist" vehicle as the one involved in the same accident with the highest impact score, excluding the user's own vehicle. For pedestrians, the striking vehicle is explicitly

identified. This allows us to calculate an `impact_delta`, representing the structural disadvantage of the user (e.g., a cyclist hit by a truck results in a high negative delta).

Location Deduplication

Contrary to the dataset description, multiple location entries were found for single accident IDs, likely due to first responders logging entries for every intersecting street. To resolve this, we implemented a `completeness_score`. We assigned weights to critical columns (Road Category: 2.0, Speed Limit: 2.0, others: 1.0). For each accident, the location entry with the highest weighted score, indicating the most data-rich description of the scene, was selected, ensuring the model trains on the highest quality data available.

3.3 Handling Missing Values (Imputation Strategy)

We distinguished between "structural missingness" (values that should not exist) and "data quality missingness" (values that are unknown).

- **Structural Missingness:** Pedestrians, by definition, do not have a vehicle category or motor type. For these cases, we explicitly imputed a value of `-1` or `'none'` to indicate "Not Applicable," preventing the model from treating these as missing data.
- **Data Quality Missingness:** For users who *should* have data (e.g., drivers) but lack it, we imputed a value of `0` or `'Unknown'`.
- **Other Vehicle Imputation:** If no opposing vehicle was involved, columns related to the "other" vehicle were set to `-1`. However, if a second vehicle ID existed but its characteristics were missing, we imputed `Unknown` to differentiate this state from single-vehicle accidents.

Finally, rows missing the target variable `injury_severity` were dropped, as they cannot be used for supervised learning.

3.4 Feature Engineering

To capture complex non-linear relationships, we generated 34 new features across four domains.

Temporal Cyclical Features

Raw timestamps are ill-suited for many models due to the discontinuity between 23:59 and 00:00. We decomposed time into cyclical components using Sine and Cosine transformations for hours, days of the week, and months. Additionally, we bucketed hours into a `time_of_day` feature (Night, Morning Rush, Midday, Evening Rush) to assist tree-based models in identifying high-level patterns.

Road Complexity Index

We hypothesized that complex road environments increase accident probability but might decrease severity due to lower speeds. We engineered a `road_complexity_index`, a composite score normalized between 0 and 10. This index aggregates weighted scores from:

- **Intersection Type:** High weights for roundabouts and multi-branch intersections.
- **Road Category:** Higher weights for urban communal ways vs. motorways.
- **Traffic Regime:** Penalties for variable assignment lanes.
- **Lane Count:** Higher complexity for multi-lane roads.

Complementing this, a binary `surface_quality_indicator` was created, set to 1 only if both the pavement condition was normal and the longitudinal profile was flat.

Vehicle and User Attributes

Vehicle categories were simplified from over 30 specific codes into 6 broad classes (Bicycle, Powered 2-3 Wheeler, Light Motor Vehicle, HGV/Truck, Bus/Coach, Other) to reduce dimensionality. For users, we transformed the `year_of_birth` into an `age` feature and further binned it into sociologically relevant `age_groups` (e.g., Child/Teen, Senior). Safety equipment flags (seatbelts, helmets, airbags) were consolidated from three separate columns into binary "Used/Not Used" indicators to resolve data sparsity.

3.5 Feature Selection & Cleaning

Following engineering, the dataset underwent a rigorous cleaning process. Invalid data, such as speed limits exceeding 130 km/h or negative values for age, were filtered out. We removed high-cardinality identifiers (IDs, address strings) and columns with excessive missingness (e.g., `width_central_reservation`) that offered little predictive value. Crucially, the raw target variable `injury_severity` (4 classes) was re-mapped to an ordinal `injury_target` (0: Uninjured, 1: Light Injury, 2: Hospitalized/Dead) to address class imbalance and better reflect the triage needs of first responders. The final dataset consists of 57 refined features ready for the modeling pipeline.

3.6 Resampling

Regarding the target value of injury severity, the dataset exhibits a clear class imbalance towards less severe cases (47% non-injured, 36% injured, and 16% heavily injured) biasing models trained towards the majority class. Given the importance of correctly assessing cases of heavy injury, resampling was applied to the dataset to remove class imbalance. Both over- (SMOTE) and undersampling were considered and tested. Given the large amount of data at disposal (over 400k) and considering compute constraints, random undersampling (130k) was ultimately chosen.

4 Data Mining

First, we partitioned the dataset into training and testing subsets, using observations from 2019 to 2022 as the training data and the year 2023 as the test set. During model selection and hyperparameter tuning, only the training data was used, while the test set was kept strictly separate to prevent any data leakage and ensure an unbiased assessment of selected models during the final evaluation.

Our primary metric for both model selection and evaluation was the macro F1-score. Since the task is an ordinal classification problem, metrics such as weighted Cohen's kappa can also provide useful insights and were considered during evaluation. However, in our use case, correctly identifying the severely injured class is particularly critical: the consequences of failing to detect a severely injured person are far more serious than those of incorrectly classifying someone as injured when they are not.

In such settings, it can be reasonable to implement a cost matrix. However, this approach is highly sensitive to the chosen cost ratios and can easily lead to an excessive emphasis on the minority class at the expense of precision. It also makes it more difficult to interpret our results. We therefore relied on macro F1 as our main metric and placed particular emphasis on the recall of the severely injured class. Because this class is highly underrepresented, using macro F1 as the optimization objective automatically increases the sensitivity to misclassifications of this class without explicitly introducing a cost matrix.

Therefore, instead of using a simple majority-vote baseline, we implemented a domain-specific baseline that relies exclusively on the speed-limit feature to predict accident severity

$$\hat{y} = \begin{cases} \text{uninjured}, & \text{if speed\_limit} \leq 50, \\ \text{injured}, & \text{if speed\_limit} < 100, \\ \text{severely injured}, & \text{otherwise.} \end{cases}$$

4.1 Models

To gain an initial understanding of the difficulty of the prediction task, we experimented with a broad range of machine learning models while applying only minimal hyperparameter tuning including models based on logistic regression (ordinal, lasso, ridge), ensembles (Random Forests, HistGradientBoosting, CatBoost) and simple neural networks. Across these experiments, we observed that all machine learning models substantially outperformed the baseline method. While originally considered, methods explicitly designed to leverage the ordinal structure of the target variable did not achieve better performance compared to other approaches in terms of both F1 and misclassifications error types, eliminating them from consideration. Applying Occam's Razor, our focus was narrowed to three models for fine-tuning and selection: Ridge Classification, Balanced Random Forest and CatBoost. While the first is Logistic Regression with applied L2-regularization and the second a random forest with rebalancing during subsampling, CatBoost is a boosting algorithm similar to XGBoost chosen for its improved support

Model	Hyperparameter	Search Space	Optimal Value
Ridge	alpha	$\{0.05, 0.10, \ldots, 10\}$	2.0
BRF	estimators	$[50, 400] \cap \mathbb{Z}$	400
	max_depth	$[3, 20] \cap \mathbb{Z}$	18
	min_samples_leaf	$[1, 20] \cap \mathbb{Z}$	1
	max_features	{`sqrt`, `log2`}	`sqrt`
	criterion	{`gini`, `entropy`}	`gini`
	sampling strategy	{`all`, `not minority`}	`all`
	replacement	{`True`, `False`}	`False`
	ccp alpha	$[10^{-6}, 0.1]$	10^{-6}
CatBoost	iterations	$[1000, 7000] \cap \mathbb{Z}$	5000
	learning rate	[0.01, 0.2]	0.0194
	depth	$[4, 10] \cap \mathbb{Z}$	10
	L2 leaf regularization	$[10^{-2}, 10]$	0.01
	border count	$[32, 255] \cap \mathbb{Z}$	32

Table 1: Search space and optimal hyperparameters for Ridge, BRF and CatBoost.

for categorical data, fitting to our data composition.

4.2 Model Selection

Considering the size of our dataset and computational constraints, model selection in terms of hyperparameter tuning was conducted for each model using 3-fold cross validation with shuffling to avoid biases from the original ordering, utilizing Bayesian Optimization and F1-Macro for the ensembles. While Ridge Regression could only be tuned regarding its regularization strength α, both ensemble methods offered more options regarding tree pruning relevant for avoiding overfitting. Table 1 shows the hyperparameters considered. Approaches selection was similarly achieved using both F1-Macro and confusion matrices, used to evaluate recall/misclassifications on (high) injury cases, with the same 3-fold cross validation.

- **Ridge Classifier:** Ridge classification provides fast and computationally efficient training and involves only a single hyperparameter to tune. Because ridge regularization tends to perform well when many features contribute small effects, it aligns well with our assumption that the predictive signal in our dataset arises from a broad set of variables rather than a few dominant predictors. In our experiments, we found it necessary to use under-sampling to compensate for the under-representation of the minority classes, as linear models are highly sensitive to class imbalance.
- **Balanced Random Forest:** Although an individual decision tree performed reasonably, our experiments showed that using an ensemble of trees significantly im-

proved predictive performance while still retaining interpretability through feature importance scores and the option to inspect individual trees. We also found that the Balanced Random Forest variant is especially useful: while macro F1 remained stable compared to the standard Random Forrest, performance on the severely injured class improved substantially, which is crucial for our application, even though slight reductions were observed for the non-injured and slightly-injured classes.

- **CatBoost:** CatBoost achieved the strongest performance in our preliminary experiments, making it a natural candidate for more in-depth analysis. An important advantage of this method is its ability to efficiently handle categorical features, which constitute the majority of our dataset. We set the auto_class_weights parameter to Balanced, so that class weights are inversely proportional to class frequencies, removing the need for additional undersampling.

4.3 Clustering

In this study, we explore four clustering algorithms to analyze accident data: K-Means, K-Prototypes, HDBSCAN, and Agglomerative Hierarchical Clustering. Each method has distinct characteristics that motivate its selection.

- **K-Prototypes:** K-Prototypes is well-suited for datasets that combine numerical and categorical variables, which makes it an appropriate choice for accident data dominated by categorical attributes. Its ability to cluster mixed-type data in a single unified framework ensures that important categorical patterns, such as road characteristics or vehicle types, are not lost while still incorporating relevant numerical information.

- **HDBSCAN:** HDBSCAN is motivated by its strength in detecting clusters of varying density and its capacity to identify noise points. Accident datasets often contain heterogeneous patterns and rare accident types; therefore, a density-based method that can naturally isolate such irregular cases without forcing them into clusters is highly desirable. Using Gower distance allows HDBSCAN to operate effectively on mixed data.

- **Hierarchical Clustering:** Agglomerative hierarchical clustering is chosen for its interpretability and flexibility. It does not require pre-specifying the number of clusters. When paired with Gower distance, it becomes a useful exploratory tool for mixed-type accident variables, helping to reveal meaningful groupings even before a final cluster solution is selected.

Parameter tuning was conducted by varying the most influential hyperparameters of each clustering algorithm. For Agglomerative Clustering and K-Prototypes, the number of clusters was set to $k = 3, 5, 7$ to explore solutions of varying granularity. For HDBSCAN, which does not require a predefined number of clusters, the *minimum cluster size* parameter was set to $k = 10, 20, 50$ to control the sensitivity of the algorithm to

dense regions in the data. These values were chosen as they provide interpretable cluster structures while remaining computationally feasible for the dataset size.
After completing the parameter tuning and selecting the final clustering configuration based on interpretability and silhouette performance, a K-Prototypes model with $k = 3$ produced a cluster label for each accident record. The resulting cluster assignments were incorporated into the dataset as an additional feature.
To derive meaningful accident profiles from the clustered dataset, we conducted a structured categorical overrepresentation analysis. For each cluster and each categorical feature, we computed the conditional distribution

$$P(X = x \mid C = c),$$

and compared it to the global distribution $P(X = x)$. A category was considered *cluster-characteristic* if it satisfied the following criteria:

- **Lift filter:** A minimum overrepresentation of

  $$\text{lift}(x, c) = \frac{P(X = x \mid C = c)}{P(X = x)} > 1.5.$$

  This ensures the category is substantially more common in the cluster than in the overall population.

- **Support filter:** The category must represent more than 3% of observations within the cluster, preventing spurious rare categories from being selected.

- **Dominance and variance filters:** Features were excluded if their distributions were nearly identical across clusters or if the same dominant category appeared in all clusters, as such features do not contribute to inter-cluster differentiation.

For numerical features, we computed summary statistics within each cluster, including the mean, median, standard deviation, and interquartile range, and compared these to the global distribution of the feature. A numerical feature was considered *cluster-characteristic* if its mean or median significantly deviated from the global average, for instance, by more than one standard deviation, indicating that the cluster exhibits unusually high or low values for that feature.

5 Evaluation

5.1 Classification

To evaluate the selected models, we followed the systematic methodology described in Section 4 and used the held-out test set from the year 2023, while the data from 2019-2022 were used for model training with the hyperparameter configurations derived above. Owing to the large size of the test set, its class distribution closely matches that of the full dataset (47% non-injured, 36% injured, and 16% heavily injured), ensuring that our evaluation is effectively stratified with respect to injury severity. We report class-wise precision, recall, and F1-scores, along with micro-F1, weighted-F1, and weighted Cohen's kappa. We also compute confusion matrices and generate precision-recall curves.

5.2 Clustering

To ensure a consistent comparison between the algorithms, all clustering results were evaluated using the silhouette score, which measures the cohesion within clusters and the separation between them. The silhouette score was computed on the preprocessed feature space used by each algorithm, ensuring methodological consistency across heterogeneous clustering approaches. A higher silhouette score indicates a better-defined cluster structure. The table below summarizes the scores attained by each method:

Algorithm	**Parameter**	**Silhouette Score**	**Largest Cluster [%]**
K-Prototypes	3	0.032305	38.8
K-Prototypes	5	0.040767	38.8
K-Prototypes	7	0.026212	38.8
Agglomerative	3	0.107160	100.0
Agglomerative	5	0.072454	100.0
Agglomerative	7	0.062951	100.0
HDBSCAN	10	0.088780	85.4
HDBSCAN	25	0.070354	85.4
HDBSCAN	50	0.048149	85.4

Table 2: Silhouette scores and largest cluster percentage for each algorithm and parameter setting.

Although Agglomerative Clustering and HDBSCAN yielded the highest silhouette scores, both methods produced highly unbalanced cluster solutions in which the vast majority of observations collapsed into a single cluster. Such degenerate solutions offer little analytical value, as they do not meaningfully differentiate between accident types. In contrast, K-Prototypes generated more balanced clusters with substantially clearer interpretability. Among the tested configurations, choosing $k = 5$ provided the best tradeoff between granularity and practical interpretability, making it the most suitable choice for deriving accident profiles.

6 Results

Our primary goal is to create a model capable of classifying injury severity to a satisfactory degree. We expect the clustering analysis to uncover specific, evidence-based "accident personas," such as low-speed urban collisions, high-velocity rural incidents, and collisions involving vulnerable road users. From the feature importance analysis, we expect to confirm that vehicle type (motorcycle vs. car) is a dominant predictor of severe injury. We also anticipate that factors like road category, lighting conditions, driver's age, and the use of safety equipment will be highly significant in predicting an individual's outcome.

6.1 Clustering

The following section presents detailed accident profiles for each cluster, summarizing the key categorical patterns and numerical deviations that distinguish them.

Cluster 0: Midday pedestrian-involved accidents with older road users

This cluster is characterized by accidents that predominantly occur around midday and often involve older or middle-aged individuals. Pedestrian involvement is more typical here than in other clusters, and the accidents tend to take place in locations frequently associated with moderate traffic density.

Cluster 1: Night-time accidents with younger to middle-aged adults

Accidents in this cluster occur mainly during nighttime and involve a broad spectrum of younger age groups, including young adults, adults, and teenagers. Pedestrian-related features indicate reduced visibility or unclear pedestrian actions, suggesting conditions with lower lighting and potentially higher uncertainty in movement patterns.

Cluster 2: Morning rush hour collisions

This cluster mostly contains accidents happening during the morning peak traffic period. Specific collision types occur more frequently here, and certain weekdays are slightly overrepresented, pointing to commuting patterns. These accidents reflect typical rush-hour dynamics with increased traffic flow.

Cluster 3: High-speed urban location collisions with defined pedestrian involvement

Cluster 3 shows a strong concentration of accidents in urban or densely built environments. Multiple collision types occur more frequently here, often involving pedestrian locations and actions typical for structured intersections or built-up areas. The cluster also exhibits a pattern of consistent seat-belt usage among vehicle occupants, suggesting regulated or lower-speed contexts.

Cluster 4: Low-speed high-frequency urban collisions with specific collision types

This cluster is dominated by accidents in certain high-frequency urban locations and is marked by a few characteristic collision types. Pedestrian involvement can occur but is less central than in other clusters. The demographic structure is skewed toward adults, and the accident patterns suggest regular urban traffic situations with typical maneuver-related collisions.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
Gemini 2.5/3 Pro	Rephrasing	Throughout	+++
Gemini 2.5/3 Pro	Code Generation and Code Debugging	Throughout	+++

Unterschrift
Mannheim, den 30. Oktober 2025