Ahead of the Impact: Predicting Injury Severity for Safer Roads

Oviya Adhan: oviya.adhan@berkeley.edu • Nory Arroyo: nory_arroyo@berkeley.edu
Caitlin Gainey: cngainey@berkeley.edu • Christine Sako: christine.sako@berkeley.edu

Abstract

With the rise of automated driving technologies, understanding factors influencing crash injury severity is vital for safer road integration. Using California crash data, we compared machine learning models to predict injury severity levels, focusing on improving detection of fatal outcomes. The Random Forest model performed best, demonstrating its strength in handling complex, imbalanced data. This work supports the development of predictive safety tools to inform future automated vehicle systems and infrastructure planning.

Introduction

As automated systems become more prevalent on the road, there is a growing need for research that informs how these technologies can be safely and equitably integrated into existing transportation ecosystems. Understanding the underlying conditions and patterns that contribute to crash severity is essential for designing systems that can effectively anticipate and respond to real-world driving scenarios. We attempt to design machine learning models that are designed to contribute to the broader research landscape by identifying patterns in historical crash data that can guide the development of predictive safety features, inform simulation scenarios, and help prioritize risk factors that automated systems must learn to recognize and navigate. This approach supports proactive infrastructure planning, policy formation, and future system design by surfacing insights that may not be obvious through rule-based engineering alone. As automated driver assistance and intelligent traffic systems evolve, our findings can help inform the scope of their deployment and the safeguards required to support their responsible use.

Related work

Previous research by Ardakani et al. (2023) has compared decision trees, random forest, multinomial logistic regression, and naive Bayes machine learning models to interpret the cause of car accidents and define solutions to minimize them. The authors reported three of the four models produced an “acceptable level of accuracy” for car accident prediction. Similarly, Ahmed et al. (2023) explores various machine learning models to predict injury severity resulting from road accidents in New Zealand. Their explored models include random forest, decision jungle, adaptive boosting, extreme gradient boosting, light gradient boosting, and categorical boosting. In addition, the researchers used explainable ML (XML) to evaluate the relationship between the severity of road accident injuries and their contributing factors. Understanding these relationships allowed the researchers to retrain and tune the models to make more accurate predictions. The study highlighted random forest to be the best performing classification model with accuracy, precision, recall, and F1 all falling within the 81-82% range.

Dataset

We use data from the California Crash Reporting System (CCRS), available from the California Open Data Portal (California Department of Technology, 2025). The dataset consists of 3 tables: crash, party, and injury - collected by the California Highway Patrol from 2016 to the current date. We focus on 2024 data for training and validation and 2025 data for testing (see **Table 1** of the Appendix) to simulate a real-world scenario where models are trained on historical data and tested on future events.

Our target variable, `ExtentofInjuryCode`, represents the most severe injury per crash and `None`, `Minor`, `Serious`, to `Fatal`. By assigning each crash a single worse-case outcome, the dataset reduces variability caused by differing numbers of people involved in each crash. This simplifies modeling and allows for a more consistent analysis of how commuting-related factors relate to injury severity.

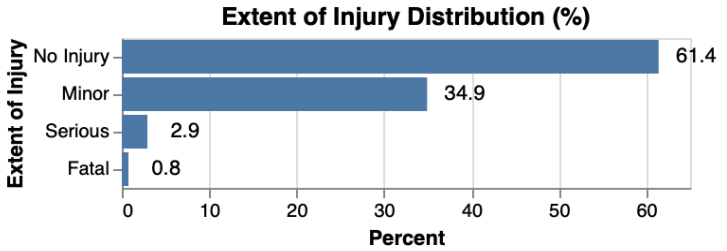
The first pre-processing step was to combine the crash, party, and injury tables together into a single dataset, separated by year (one for 2024 and one for 2025). We selected features based on missing value thresholds and what we thought would be most relevant to influence injury per the CCRS variable definitions (see **Variable Definitions** of the Appendix). Resulting were 5 features from the `crashes_*` and 6 from `parties_*`. All chosen features are categorical with the exception of `SpeedLimit` (float) and `IsHighwayRelated` (boolean). From these selected features, we filtered out rows with NaN values in feature columns and converted NaN values in `ExtentOfInjuryCode` to `No Injury`.

Our main preprocessing challenge was consolidating multiple party and injury records linked to the same crash into a single crash-level record. In the consolidated dataset, we aggregated values from records sharing the same `CollisionId` into arrays for each feature. Exceptions included `SpeedLimit` and the target variable `ExtentOfInjuryCode`, for which we retained only the maximum value per crash. For `ExtentOfInjuryCode`, this meant selecting the highest severity level based on our custom ordering.

After data cleaning, we split data into training, validation, and test sets and conducted EDA on the training dataset, detailed below. The post-processing dataset shapes were: Train: (218338, 12), Val: (54585, 12), Test: (103669, 12).

Our first EDA step was to understand the distribution of our target variable, `ExtentOfInjuryCode`. As observed, `No Injury` makes up 61.4%, `Minor` 34.9%, `Serious` 2.9%, and `Fatal' 0.8% of the total samples in the training dataset, indicating severe class-imbalance that will be addressed during modeling. See Fig 1 below:

Fig 1. Distribution of Outcome Variable



Next, we looked at the distribution of categorical variables across each injury extent level (Figs. 3–14). Highlights:

- `**Weather1**`: The majority of fatal crashes (~10%) occurred in unknown weather.
- `**LightingDescription**`: Daylight conditions were most common for crashes overall. However, the highest percentage of fatal crashes (~3.2%) occurred on dark streets without functioning lights.
- `**CollisionTypeDescription**`: Vehicle-pedestrian collisions had the highest percentage for actual injuries: `Minor` at 57.7%, `Serious` at 18.6% and `Fatal` at 11.4%
- `**IsHighwayRelated**`: Crashes on highways showed a slight decrease for `Serious` and `Fatal` crashes, and a decrease from 38.7% to 32% for `Minor `injuries.
- `**AirbagDescription**`: Airbag deployment was more frequent in crashes involving `Serious` or `Fatal` injuries.

`**SpeedLimit**` had similar distributions across all injury levels except `Fatal` crashes, which were denser in the 2nd quartile (Fig. 15), suggesting they more often occur at higher speed limits. Based on this EDA, we normalized numerical variables using training set statistics, one-hot encoded categorical features, and used the resulting files for modeling.

Methods

To predict injury severity in traffic crashes, we implemented and compared four supervised learning algorithms: a Multiclass Logistic Regression model (baseline), a Neural Network, a Random Forest model, and an XGBoost model.

Baseline: Multiclass Logistic Regression

For the baseline model, we used a multiclass logistic regression classifier with balanced class weights. Multiclass logistic regression is a statistical model used to classify data into different categories by calculating log-odds of class membership as a function of the input features. To assign data to a class, the model uses a softmax function to convert raw scores into a normalized probability distribution across all classes. Given its simplicity, interpretability, and computational efficiency, it is commonly used as a machine learning algorithm baseline. Using this model in conjunction with balanced class weights allowed the model to account for substantial class imbalance within our dataset. Though the model was limited in its ability to capture nonlinear relationships or interactions among features, the multiclass logistic regression classifier provided a reasonably well-performing benchmark for our study.

Improvement 1: Neural Network

For the first attempted improvement over the baseline model, we implemented a neural network using TensorFlow. Neural networks utilize layers of interconnected neurons, where each neuron applies a weighted sum followed by a nonlinear activation function, to learn complex patterns in the data. The network learns by minimizing a loss function such as categorical cross-entropy via an optimizer, which updates the weights of the network connections to reflect the patterns in

the training data. Given its ability to capture nonlinear relationships and interactions from input features, neural networks form a comprehensive view of the factors that contribute to injury in a car crash. Thus, the neural network model was an appropriate candidate for an improvement over the baseline. We optimized our network architecture to perform multiclass classification by testing activation-optimizer pairs with different layer amounts and sizes with a softmax output layer.

Improvement 2: Random Forest Classifier

The next improvement over the baseline was a random forest classifier, which is an algorithm well-suited for tabular datasets with many categorical features. Random forests are ensembles of decision trees, where each tree is trained on a random subset of the data and features. In a multiclass prediction problem, each tree will provide a class vote, and the majority vote becomes the final classification of the datapoint. One strength of the random forest classifier is its inherent feature selection process, where each tree chooses to split based on how informative a feature is. This makes the model robust to multicollinearity and less relevant features. Additionally, the model can learn complex interactions between variables. To improve generalization and model performance, we used a range of techniques including stratified k-fold cross-validation during training. This ensures that the model performs consistently across different subsets of the training data rather than relying on a single train-validation split. We included a PCA transformation in the hyperparameter grid to reduce dimensionality and prevent overfitting, and integrated SMOTE (synthetic minority oversampling technique) in the training pipeline to address class imbalance by synthetically generating more minority class samples.

Improvement 3: XGBoost Classifier

The final improvement over the baseline was an XGBoost classifier. XGBoost is a gradient boosting algorithm that builds trees sequentially, where each new tree attempts to correct errors made by the ensemble of previous trees. Unlike random forests, XGBoost learns additive models rather than an average of independent trees. The XGBoost method is known for its high performance on structured data, as it learns from its errors, and its inclusion of regularization to prevent overfitting, making it a good candidate to outperform the baseline. To maximize model performance, we conducted a randomized hyperparameter search over a grid that included variations in learning rate, number of estimators, and L1/L2 regularization terms. This model architecture also included k-fold cross-validation in the training process.

Experiments

To optimize our models, we ran each non-baseline model through a series of experiments, focusing on optimizing the F1-score of each model, particularly for the ‘Fatal’ class. This aligns with our ultimate goal of predicting life-endangering scenarios by balancing precision and recall. We additionally calculated accuracy to track our models’ generalization capabilities. Our Multiclass Logistic Regression baseline model utilized the scikit-learn Logistic Regression classifier and did not attempt to optimize it in order to provide a simple baseline to work on top of.

The base of our Neural Network came from the TensorFlow Keras Sequential model, configured either with a customizable number of hidden layers or as a multiclass regression when no hidden layers were specified. Experimentation was done through an ablation study with the output set to softmax. The first round tested various combinations between ReLU or Tanh activations and SGD or Adam optimizers while keeping the hidden layers to 0. Since different activation functions showed no significant accuracy differences and Adam optimizer caused some underfitting, we chose the ReLU activation with SGD optimizer. The second round tested different hidden layer setups: a single layer with 128 units and two layers with 256 and 128 units. The single-layer model performed best on generalization and F1-score while requiring less computation, so it was selected to prevent overfitting.

We trained our Random Forest classifier using a variety of tactics to improve the macro-averaged F1-score and average accuracy of the classifier, specifically focusing on the F1-score for fatal injuries. First, we used k-fold cross validation and optionally included PCA to reduce dimensionality in the hyperparameter tuning to simplify the feature inputs and improve generalization. To account for class imbalance, we implemented SMOTE in the pipeline to oversample minority classes. We used a randomized grid search to explore a range of hyperparameters, including the number of estimators, tree depth, minimum samples for splits and leaves, and PCA inclusion. For each combination, we evaluated the model using 3-fold stratified cross-validation and selected the best configuration based on the selected evaluation metrics. The optimal model did not use PCA, utilized 3-fold Stratified Cross validation, had 190 estimator trees, 5 splits, 1 minimum sample per leaf, and maximum tree depth of 30 levels.

In our first iteration using the XGBoost Classifier, we used the default hyperparameter values provided by the XGBoost library. Following the initial results, we proceeded with hyperparameter tuning by creating a grid of ranges for

parameter values such as learning rate, number of trees, and L1 and L2 regularization. We used randomized search to explore 25 random parameter combinations within the search grid, applying 10-fold cross-validation. This resulted in fitting the model 250 times. After tuning, accuracy improved slightly, while F1 metrics remained largely unchanged. The optimal model had a 0.1 learning rate, 200 estimator trees, 0.1 L1 lasso regularization, and a 1.3 L2 ridge regularization

Results

To compare our optimized models, accuracy and F1-score were evaluated on the test dataset (2025 crash data) to assess each model's generalization to unseen cases. See **Table 2** below for results.

Table 2. Model Metrics on Test Dataset

		1 - Multiclass Logistic Regression	2 - Neural Network	3 - Random Forest	4 - XGBoost
Accuracy		0.61	0.60	**0.66 ***	0.63
Macro-Averaged F1		0.37	0.37	**0.42 ***	0.40
SUBGROUP	**F1 - No Injury**	0.75	0.73	**0.76 ***	0.75
	F1 - Minor	0.48	0.49	**0.55 ***	0.52
	F1 - Serious	0.17	0.17	**0.19 ***	**0.19 ***
	F1 - Fatal	0.09	0.08	**0.17 ***	0.13

* Highest score across models in metric category

Discussion

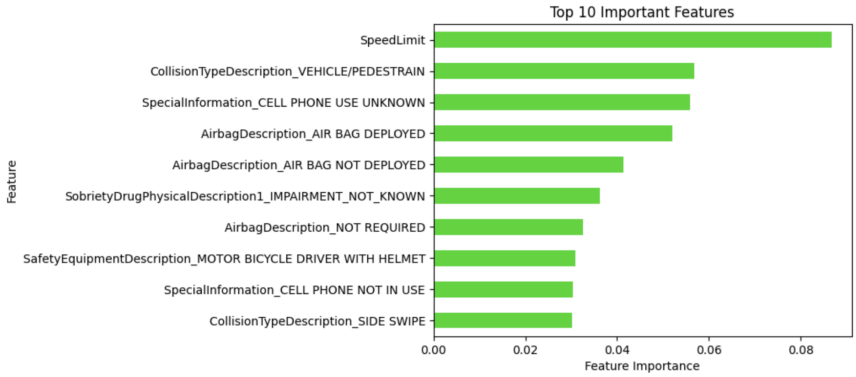
Our ultimate goal was to improve the F1-score of our `Fatal` class, but evaluate other metrics to provide a fuller picture. Focusing on the F1-score for the `Fatal` class, model performance ranked as follows: Random Forest, XGBoost, Multiclass Logistic Regression, and Neural Network, with Random Forest outperforming the next best model by 0.04. At the macro level, the macro-averaged F1-score, which equally weights all classes, the models ranked as follows: Random Forest, XGBoost, then Neural Network and Multiclass Logistic Regression tied for last. Random Forest scored a modest but clear 0.02 higher than XGBoost. Similarly, for accuracy, the models ranked the same as for the `Fatal` F1-score, with Random Forest outperforming XGBoost by 0.03. Overall, the Random Forest classifier outperformed the others across most metrics, except for the `Serious` class F1-score, where XGBoost performed equally well, making it a close second. The baseline Multiclass Logistic Regression ranked third, with the Neural Network slightly behind.

We expected the Neural Network to perform similarly to the baseline because the high dimensionality and noise from dummy variable encoding caused it to model unnecessarily complex nonlinear relationships. In contrast, logistic regression handles high collinearity between variables more robustly, even without extensive tuning. XGBoost and Random Forest both outperformed the other models, likely because their use of decision trees enables them to capture feature interactions, which is an advantage when dealing with noisy data. Random Forest makes decisions based on majority voting across independently grown trees, while XGBoost builds trees sequentially, where each tree learns from the errors of the previous ones. Although XGBoost can theoretically fine-tune itself better, it is more prone to overfitting noise due to its additive corrections. This robustness to noise likely explains why Random Forest emerged as the strongest model.

Across all experiments, improvements in accuracy and F1-scores were modest and insufficient for production-ready performance. Notably, the `Fatal` and `Serious` classes, which were critical for our goals, had significantly lower F1-scores than the `No Injury` and `Minor` classes. The largest gap between `No Injury` and `Fatal` F1-scores was 0.66 for the baseline model, while Random Forest reduced this gap to 0.59, making it the most balanced model across subgroups. This improved balance likely stems from the use of SMOTE to address class imbalance during training. Given its superior combined `Fatal` F1-score, macro-average F1-score, accuracy, and balanced class performance, Random Forest is our optimal model.

After determining our top model, we examined the top 10 most important features identified by our Random Forest model, see **Fig 2**. below:

Fig 2. Top 10 Most Important Features



As shown above, speed limit is the most important predictor of injury severity, which is expected given that higher speed limits are typically associated with more forceful impacts, and consequently, more severe injuries. Next in the ranking, we see vehicle/pedestrian collisions, which are inherently more dangerous due to the vulnerability of pedestrians. The model clearly recognizes that these types of crashes carry a higher risk of serious or fatal outcomes. Features like "Cell Phone Use Unknown" and "Impairment Not Known" also appear to be critical features in the model. This could indicate that uncertainty or missing data corresponds to chaotic or severe crashes where investigations were not possible. We also see several airbag-related features. These could help assess the severity of the impact and whether safety equipment were engaged. Overall, the Random Forest model leverages a mix of crash dynamics and even data uncertainty to assess injury severity, which gives us valuable insight into both model behavior and real-world risk factors for injury.

Conclusion

In this project, we analyzed automobile crash data from the California Crash Reporting System (CCRS) to identify patterns related to injury severity and contribute to research on safety measures for emerging vehicle technologies. Using three datasets, we examined key features, such as road condition, weather, lighting, and collision type, and modeled injury severity across four levels: No Injury, Minor Injury, Serious Injury, and Fatal Injury. Our baseline was a multiclass logistic regression, complemented by a Neural Network, Random Forest, and XGBoost classifiers. Due to significant class imbalance, we evaluated models using both accuracy and macro-averaged F1 scores, with particular focus on improving the F1 score for the Fatal Injury class to better detect life-critical outcomes. The Random Forest classifier performed best overall, likely due to its robustness against noise through bagging. It achieved 66% accuracy, a 0.42 macro F1-score, and a 0.17 F1-score for the Fatal Injury class. Future work could include reframing the problem as binary classification (`Serious`/`Fatal` vs. `Minor`/`No Injury`), prioritizing feature selection earlier to reduce noise, and leveraging greater computational resources to enhance model tuning, such as increasing the number of trees in ensemble models.

Contributions

C.S. found the datasets and maintained the project Github. She wrote the Introduction, Related Work, and Dataset sections. Instructor feedback was addressed by her. She was responsible for Phase 1 in the pre-processing stage and the baseline model in Phase 2. Presentation slides were created and organized by her. Jupyter notebooks associated with these contributions are `processing_2024.ipynb`, `processing_2025.ipynb`and `baseline_model` in our project Github repo.

O.A. ran parts of the pre-processing including handling NaN values, normalizing numeric features, and splitting the data into train, validation, and test sets. She conducted the initial exploratory data analysis as well as the interpretation of the results. In Phase 2, the neural network model was created by her and the experiments, results, and model comparison portion of the discussion sections were written by her. The Jupyter notebooks associated with these contributions are 'EDA.ipynb', 'final_processing.ipynb', and 'neural_network.ipynb' in the project repository.

C.G. wrote the methodology section and contributed to the writing of the dataset section and the feature importance portion of the discussion section. In addition, she unraveled the data in the EDA phase. Her main contribution was building and tuning the random forest classifier model and generating the resulting feature importance plot. Her code for the EDA and the rf classifier model is in the 'RandomForestClassifier.ipynb' notebook.

N.A. Contributed writing towards the data section, wrote the conclusion, and helped with the experiments portion of the paper. Additionally, she created new graphs for the EDA as per feedback granted by the instructor. For modelling, she contributed the XGBoost Model. Her work is in the revised_eda.ipynb and XGBoost_Model.ipynb notebooks.

Appendix

Table 1. Features and Outcome Variable Selection

1. `crashes_2024`, 194.5 MB, shape: (410348, 73)
   `crashes_2025`, 65.9 MB, shape: (140311, 73)
   a. Contains automobile crash data (Primary Key: `CollisionId`)
   b. **Features** selected and their datatypes after pre-processing are:
      i. `CollisionTypeDescription`, *category*
      ii. `IsHighwayRelated`, *bool*
      iii. `Weather1`, *category*
      iv. `RoadCondition1`, *category*
      v. `LightingDescription`, *category*

2. `parties_2024`, 181.3 MB, shape: (801856, 38)
   `parties_2025`, 63 MB, shape: (272242, 38)
   a. Contains data from parties involved with crashes (Primary Key: `PartyId`, Foreign Key: `CollisionId`)
   b. **Features** selected and their datatypes after pre-processing are:
      i. `SpeedLimit`, *float64*
      ii. `MovementPrecCollDescription`, *category*
      iii. `AirbagDescription`, *category*
      iv. `SafetyEquipmentDescription`, *category*
      v. `SobrietyDrugPhysicalDescription1`, *category*
      vi. `SpecialInformation`, *category*

3. `injuredwitnesspassengers_2024`, 63.1 MB, shape: (485031, 21)
   `injuredwitnesspassengers_2025`, 24.1 MB, shape: (164580, 21)
   a. Contains data from individuals who may or may not have been injured as either witnesses or passengers involved with crashes (Primary Key: InjuredWitPassId, Foreign Key: CollisionId)
   b. Our **target variable** is:
      i. `ExtentOfInjuryCode`, category

Table 3. Baseline Logistic Regression Results

```
The training acccuracy is: 0.6134
The full training classification report is:
              precision    recall  f1-score   support

   No Injury       0.79      0.72      0.75    134059
       Minor       0.56      0.44      0.49     76242
     Serious       0.11      0.46      0.18      6385
       Fatal       0.12      0.73      0.20      1652

    accuracy                           0.61    218338
   macro avg       0.39      0.59      0.41    218338
weighted avg       0.68      0.61      0.64    218338


-------

The validation acccuracy is: 0.6122
The full validation classification report is:
              precision    recall  f1-score   support

   No Injury       0.79      0.72      0.75     33637
       Minor       0.55      0.44      0.49     18986
     Serious       0.11      0.45      0.18      1562
       Fatal       0.11      0.71      0.19       400

    accuracy                           0.61     54585
   macro avg       0.39      0.58      0.40     54585
weighted avg       0.68      0.61      0.64     54585


-------

The test accuracy is: 0.6110
The full test classification report is:
              precision    recall  f1-score   support

   No Injury       0.80      0.71      0.75     65530
       Minor       0.54      0.44      0.48     35123
     Serious       0.10      0.47      0.17      2699
       Fatal       0.05      0.68      0.09       317

    accuracy                           0.61    103669
   macro avg       0.37      0.57      0.37    103669
weighted avg       0.69      0.61      0.64    103669
```

Fig 3. ExtentofInjuryCode vs. Weather1

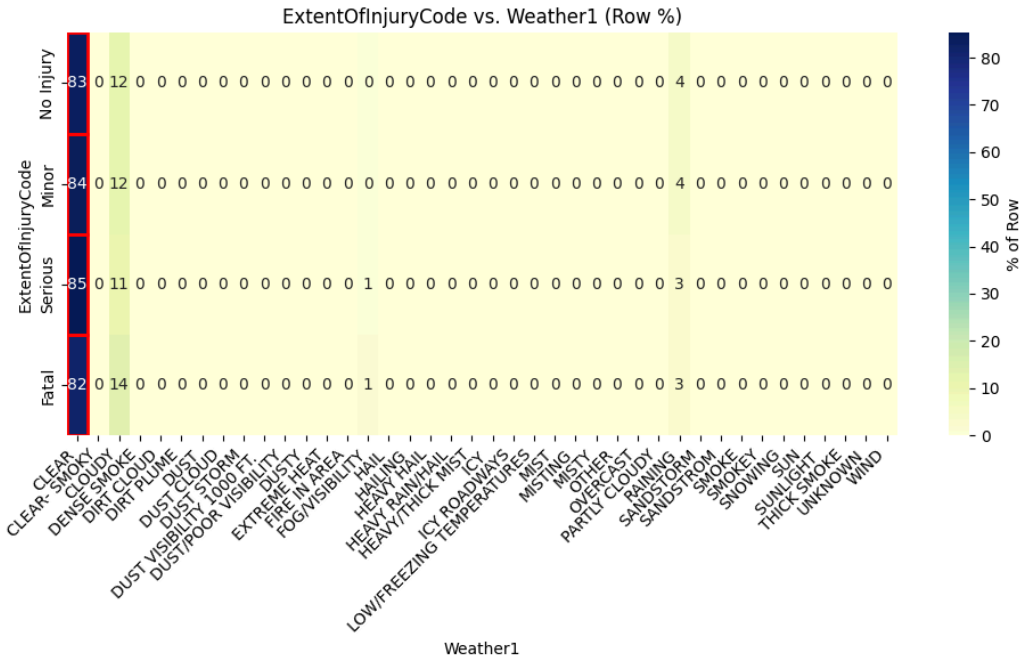


Fig 4. ExtentofInjuryCode vs LightingDescription

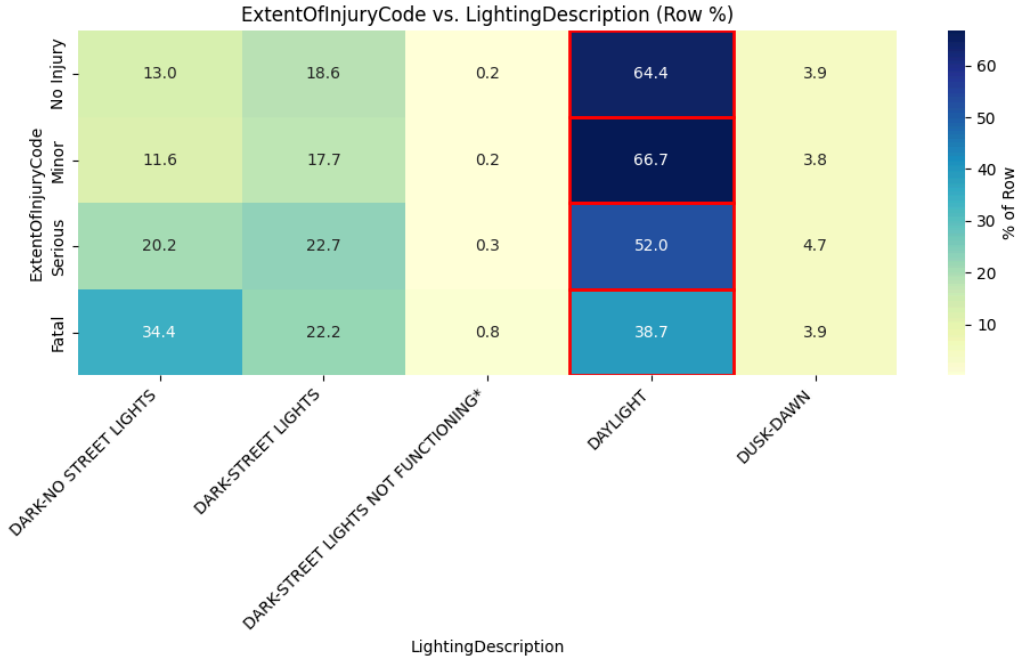


Fig 5. ExtentofInjuryCode vs CollisionTypeDescription

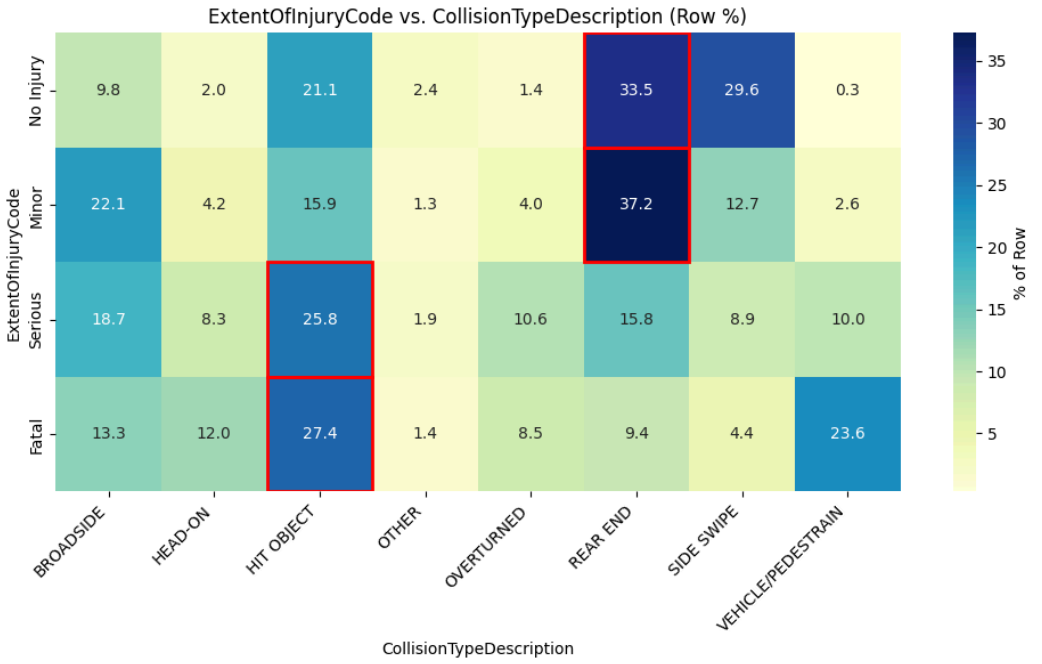


Fig 6. ExtentofInjuryCode vs IsHighwayRelated

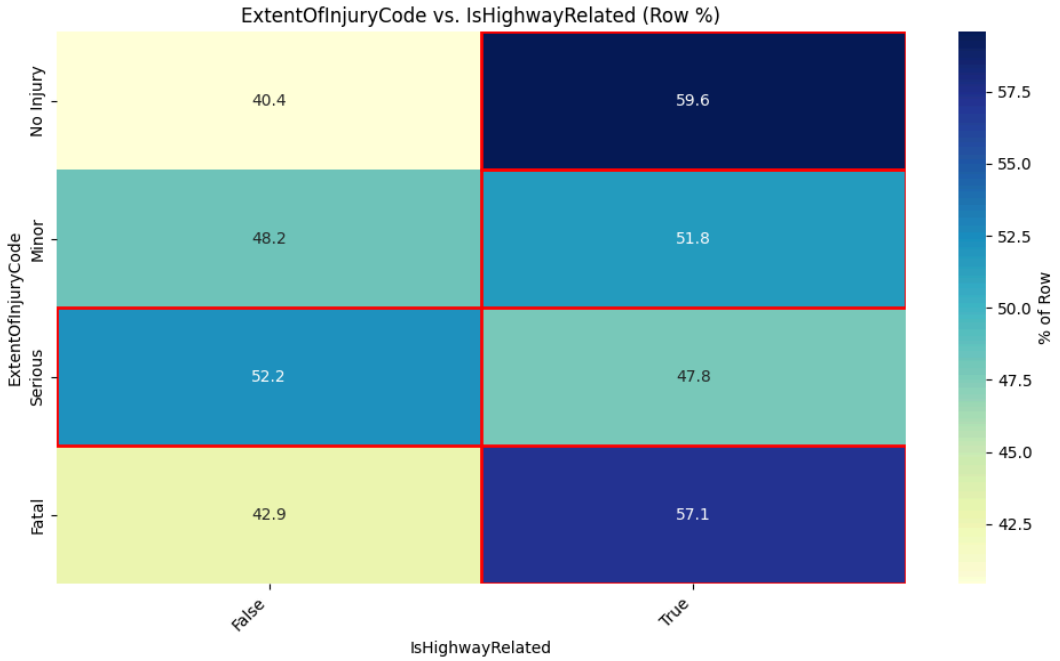


Fig 7. CollisionTypeDescription by Extent of Injury

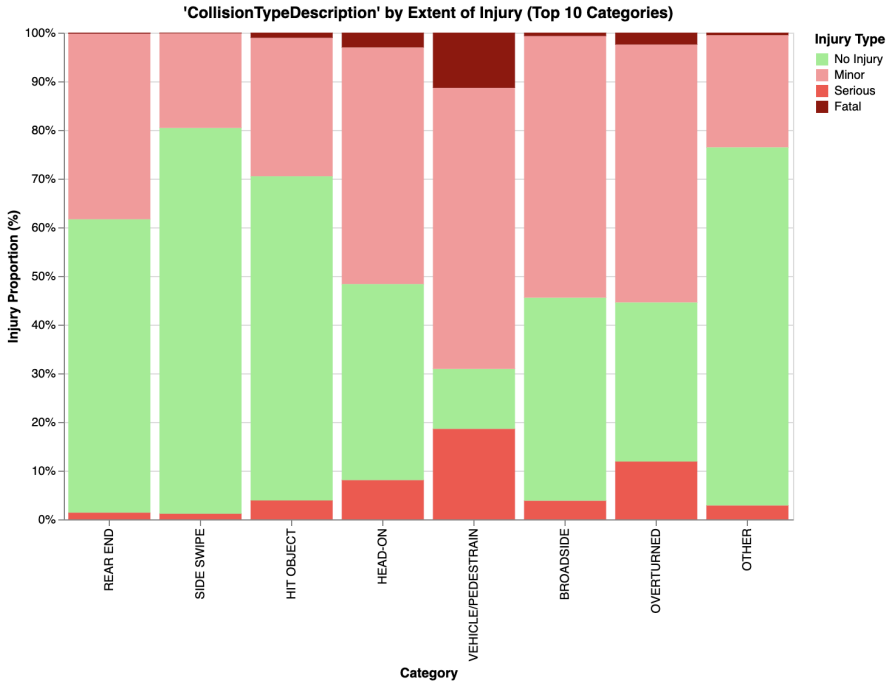


Fig 8. IsHighwayRelated by Extent of Injury

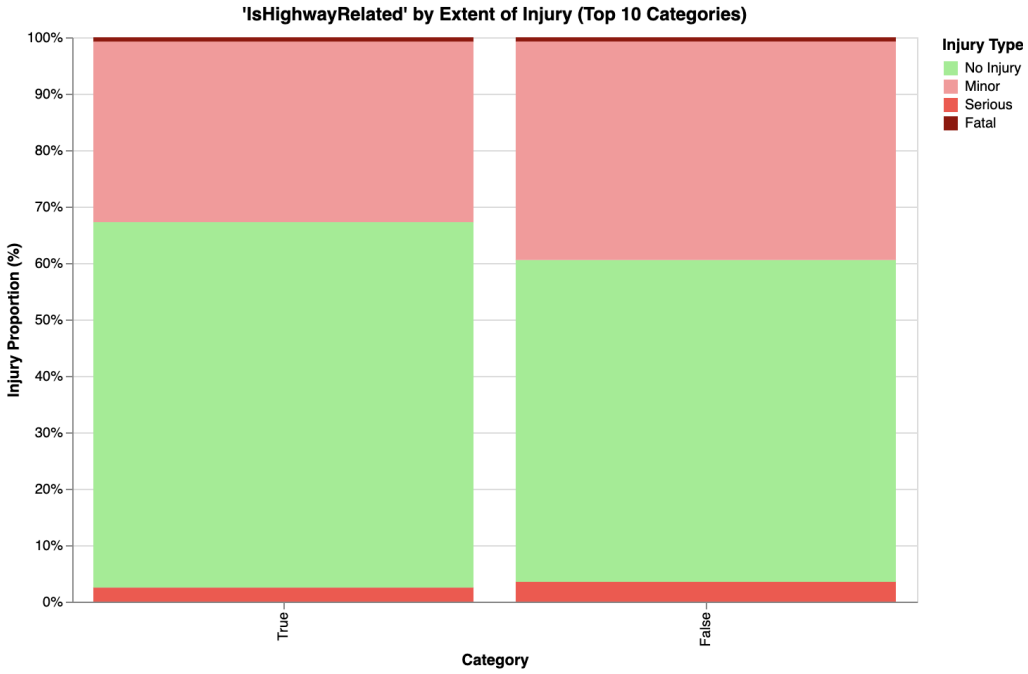


Fig 9. Weather1 by Extent of Injury

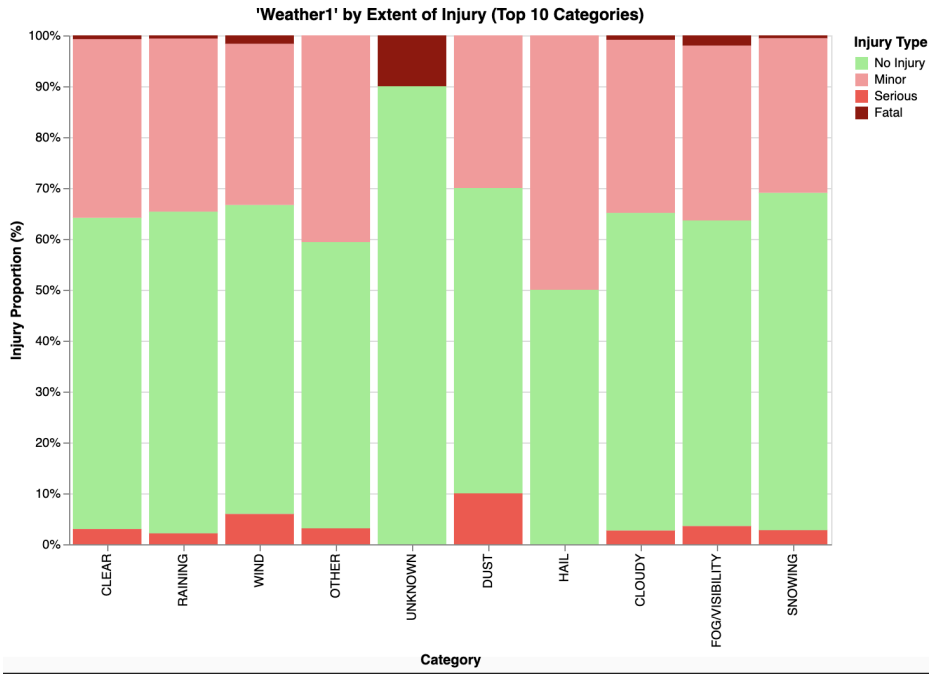


Fig 10. RoadCondition1 by Extent of Injury

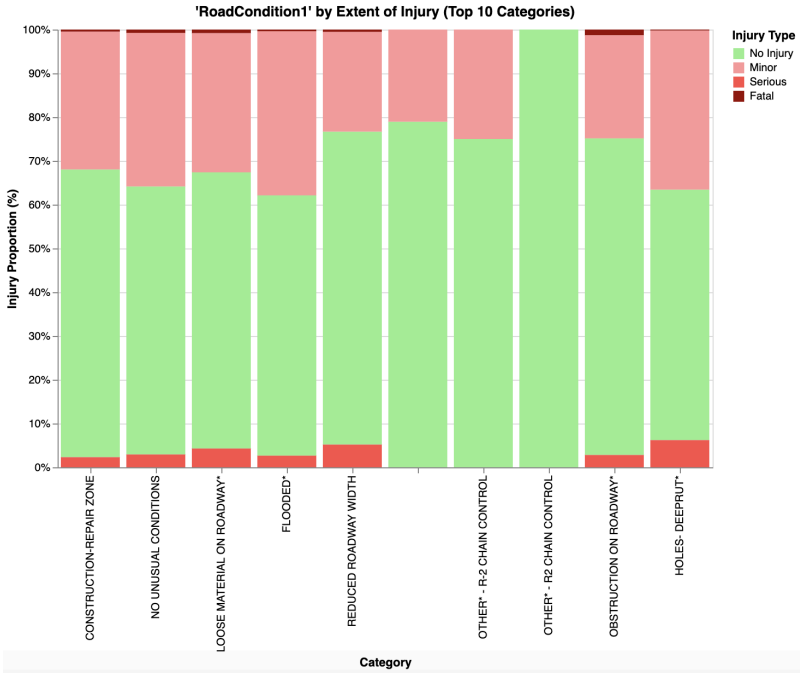


Fig 11 LightingDescription by Extent of Injury

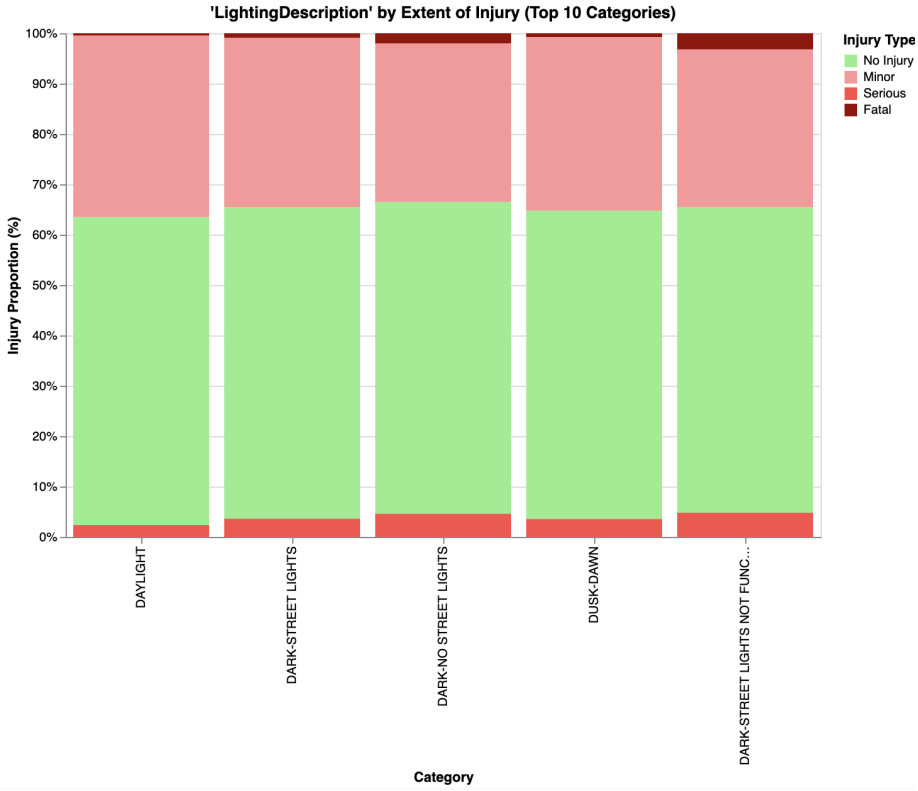


Fig 12. AirbagDescription by Extent of Injury

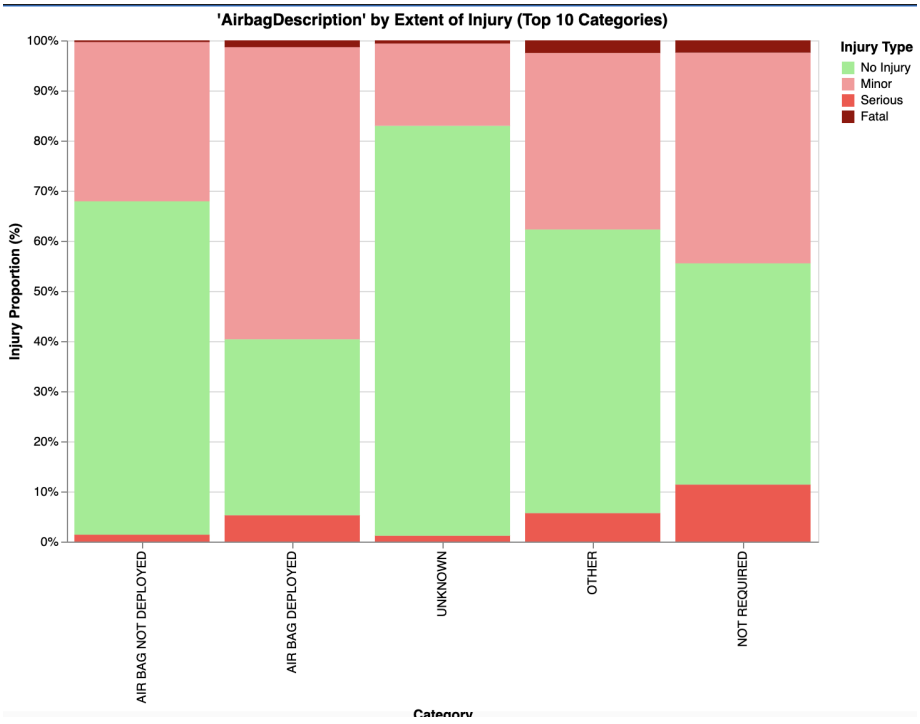


Fig 11. SafetyEquipmentDescription by Extent of Injury

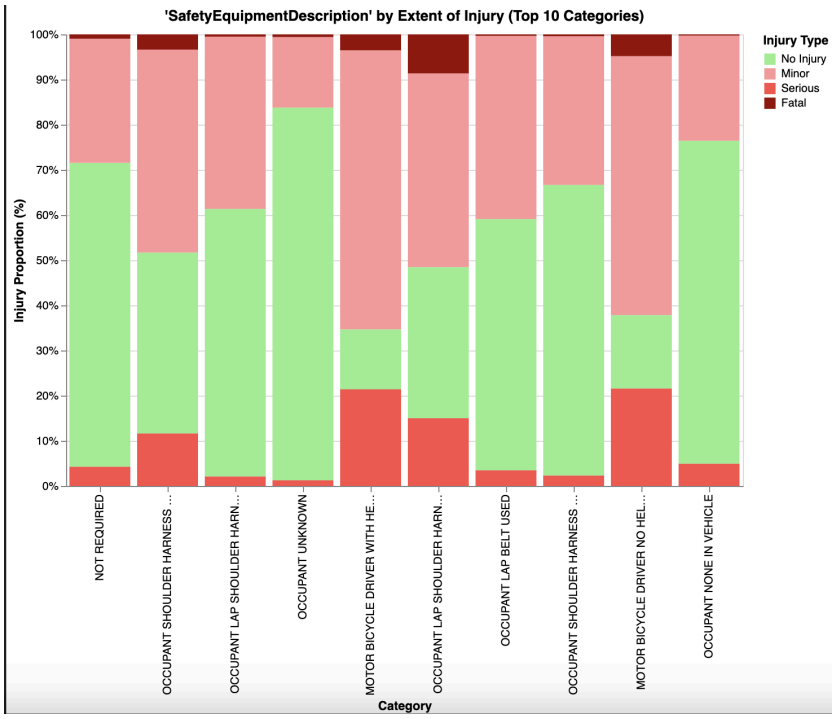


Fig 13. SobrietyDrugPhysicalDescription1 by Extent of Injury

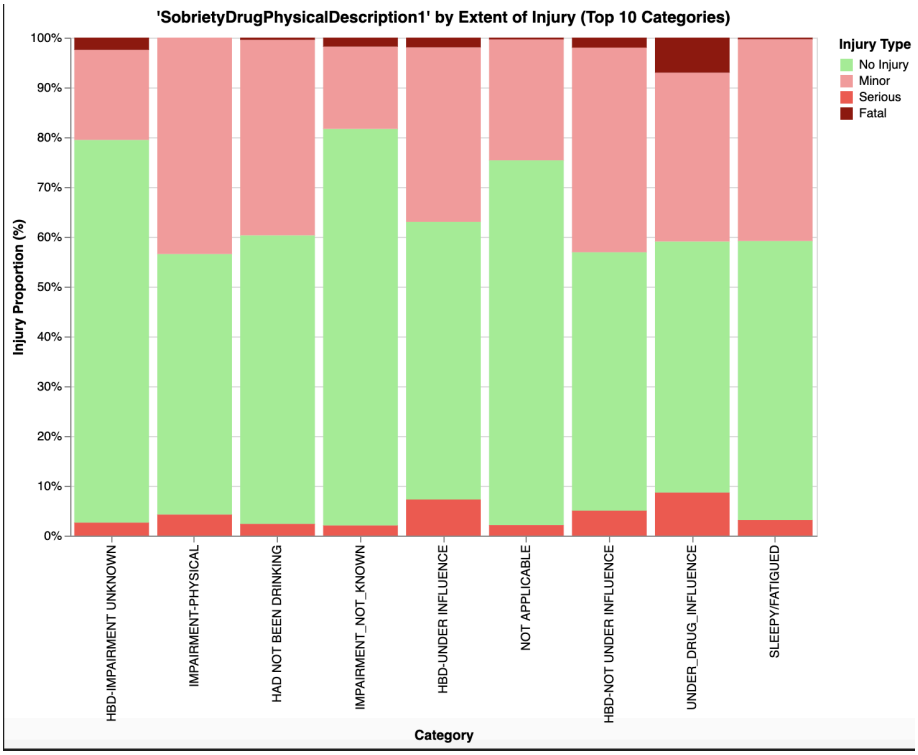


Fig 14. SpecialInformation by Extent of Injury

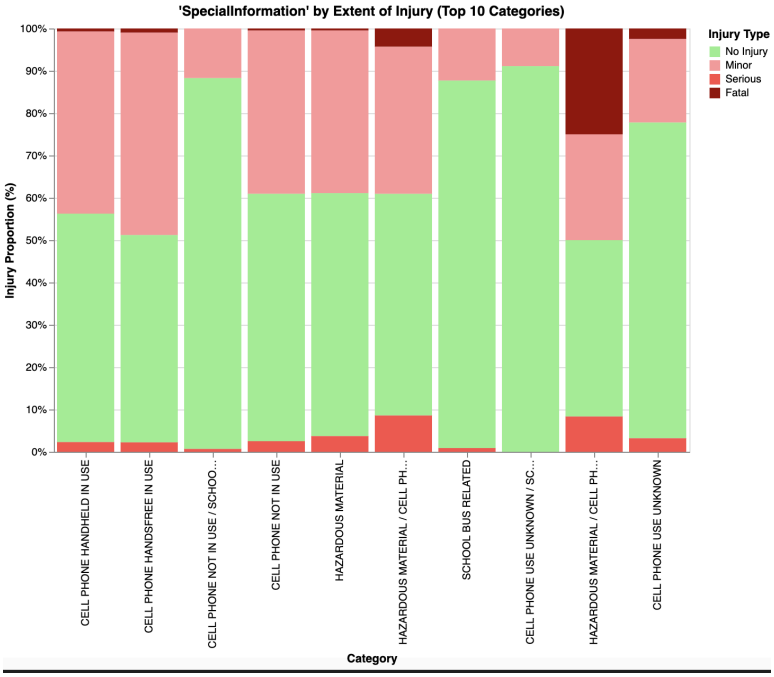
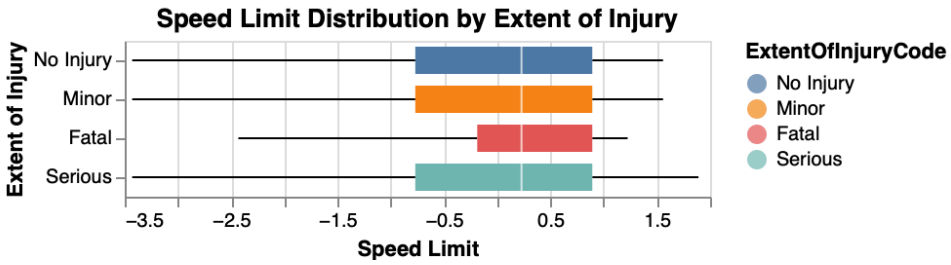


Fig 15. Speed Limit Distribution by Extent of Injury



Variable Definitions (as reported by CCRS):

`crashes_202*`:

- `***CollisionTypeDescription***`, *category:* This variable defines the type of crash. The set of descriptions is A Head-on, B Side swipe, C Rear end, D Broadside, E Hit object, F Overturned, G vehicle/pedestrian, and H Other.
- `***IsHighwayRelated***`, *bool*: This variable defines if the crash occurred on a highway or not. Potential values are True or False.
- `***Weather1***`, *category*: This variable describes the weather at the time of the crash. The set of possible weather conditions is A Clear, B Cloudy, C Raining, D Snowing, E Fog, F Other, and G wind.
- `***RoadCondition1***`, *category*: This variable describes the roadway condition at the time of the crash for the traffic lanes involved. The set of possible road conditions is A Holes or Deep Ruts, B Loose material on Roadway, C Obstruction on roadway, D Construction or repair zone, E Reduced roadway width, F Flooded, G Other, H No unusual condition.
- `***LightingDescription***`, *category*: This variable describes the lighting during the time of the crash. The set of possible lighting descriptions is A Daylight, B Dusk-Dawn, C Dark-street lights, D Dark-no street lights, E Dark-street lights not functioning.

`parties_202*`:

- `***SpeedLimit***`, *float64:* This variable defines the speed limit in the zone of the crash.
- `***MovementPrecCollDescription***`, *category*: This variable defines the movement that caused the crash. The possible options are A Stopped, B Proceeding straight, C Ran off road, D Making right turn, E Making left turn, F Making U-turn, G Backing, H Slowing/stopping, I Passing other vehicle, J Changing lanes, K Parking maneuver, L Entering traffic, M Other unsafe turning, N Crossed into opposing lane, O Parked, Q Traveling wrong way, R Other, Null not stated.

- `***AirbagDescription***`, *category*: This variable defines the state of the airbag during the time of the crash. The set of possible values is B Unknown, L Airbag deployed, M Airbag not deployed, N Other, P Not required.
- `***SafetyEquipmentDescription***`, *category:* This variable describes the safety equipment presence and use in the crash. The possible values are A None in vehicle, B Unknown, C Lap belt used, D Lap belt not used, E Shoulder harness used, F Shoulder harness not used, G Lap/shoulder harness used, H Lap/shoulder harness not used, J Passive restraint used, K Passive restraint not used, L Airbag deployed, M Airbag not deployed, N Other, P Not required, Q Child restraint in vehicle used, R Child restraint in vehicle not used, S Child restraint in vehicle with use unknown, T Child restraint in vehicle with improper use, U No child restraint in vehicle, V Driver with motorcycle helmet not used, W Driver with motorcycle helmet used, X Passenger with motorcycle helmet not used, Y Passenger with motorcycle helmet used, Null Not stated.
- `***SobrietyDrugPhysicalDescription1***`, *category:* This variable describes the sobriety of the driver who initiated the crash. The set of possible levels of sobriety is A - Had not been drinking, B - Had been drinking, under influence, C - Had been drinking, not under influence, D - Had been drinking, impairment unknown, G - Impairment unknown, H - Not applicable, Null - Not stated.
- `***SpecialInformation***`, *category*: This variable determines if other factors were involved in the collision. This includes A Hazardous materials, B Cell phone in use, C Cell phone not in use, D No cell phone/unknown, 1 Cell phone handheld in use, 2 Cell phone handsfree in use, 3 Cell phone not in use, 4 Cell phone use known, E School bus related.

`injuredwitnesspassengers_*`:

- `***ExtentOfInjuryCode***`, *category*: This variable determines the extent of the injury withstood during the collision. The possible values are Fatal injury, Suspected serious injury, Suspected minor injury, and Possible injury.

Github Repo
A link to our Github repository can be found here:
https://github.com/christinesako-berk/ds_207_final_project

Resources

Ahmed, S., Hossain, M. A., Ray, S. K., Bhuiyan, M. M. I., & Sabuj, S. R. (2023). *A study on road accident prediction and contributing factors using explainable machine learning models: Analysis and performance*. Transportation Research Interdisciplinary Perspectives, 19, Article 100814.
https://doi.org/10.1016/j.trip.2023.100814

California Department of Technology, California Department of Motor Vehicles, & California Highway Patrol. (2025). California Crash Report System (CCRS). Data.ca.gov. Retrieved June 16, 2025, from https://data.ca.gov/dataset/ccrs

Pourroostaei Ardakani, S., Liang, X., Mengistu, K. T., So, R. S., Wei, X., He, B., & Cheshmehzangi, A. (2023). Road Car Accident Prediction Using a Machine‐Learning‐Enabled Data Analysis. Sustainability (Switzerland), 15(7), Article 5939.
https://doi.org/10.3390/su15075939