Collision Severity Prediction

CPEN 355 - Final Project Report

Group 7:

Dominic Bongiorno (43068626)

Maddy Paulson (36440824)

Jake Rubin (86732823)

Public Github Repository: https://github.com/maddypaulson/group7

Backup Link to Colab:

https://colab.research.google.com/drive/1R9P3gznW7ZJOaWTcF3ZojpJUgRjyf1tg?usp=sharing

ABSTRACT

Machine Learning (ML) has become a powerful tool in analyzing complex datasets with numerous parameters, enabling the generation of accurate predictions from large samples. Its applications rapidly expand across diverse fields, including medicine, customer service, and automation. This report examines the performance of a gradient-boosting ML model designed to predict the severity of vehicle collisions. The model utilizes environmental factors such as time of day, season, weather, road conditions, and vehicle-specific details like type and year of manufacture to assess its predictive accuracy and potential applications.

This investigation concluded that a gradient boosting tree model (optimized with 50 estimators, a maximum depth of 7, and a learning rate of 0.094) could accurately determine the number of injuries from an accident given these environmental and vehicle-specific details, with a root mean squared error of approximately 0.58. However, the result for predicting fatalities was less accurate than the above model. A gradient boosting tree model (optimized with 100

estimators, a maximum depth of 7, and a restricted learning rate of 0.031) was predicted with a root mean squared error of 0.1. Despite this error being much less than 0.58, this was significantly higher than the mean of all test values, showing its lack of effectiveness.

INTRODUCTION

In 2022 alone, Transport Canada reported 1,931 fatal collisions, each resulting in at least one death[1]. This marks a 6.0% increase from the previous year and demonstrates the need for a better understanding of the factors that influence the severity of motor vehicle collisions to work towards more effective preventative measures. Due to the many factors affecting a collision, pinpointing where to employ preventative measures is extremely difficult. However, with advancements in machine learning models, the predictive abilities of ML can be utilized to analyze these factors better and predict severity. Collision severity modelling is a popular research topic, focusing on using predictions to assess the effectiveness of the models[2,3]. However, employing machine learning models to predict collision severity based on input factors like vehicle number, road conditions, and time of day is less researched and presents a gap in the literature, warranting further exploration [2,4,5].

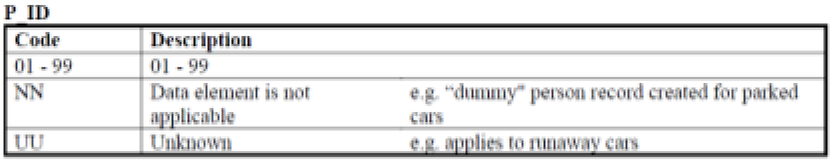
The problem described above suggests characteristics of a regression problem, which is why the collision severity prediction uses a Gradient Boosting Regression model. The model treats injuries and fatalities as separate outputs to predict and uses XGBoost for the implementation.

DATA

Data was acquired from the Canadian Open Government's National Collision Database [1]. Data from 2016-19 was used, providing approximately a million data points after cleaning.

Collisions were given a case number, with varying rows depending on the number of people involved in the collision. There were 22 columns in the dataset describing all the data collected from the collision. Some were irrelevant and could be removed immediately as they had no perceived relevance to the collision outcome. These included passenger sex, passenger age and position of the passenger in the vehicle.

The rest were more critical and had to be cleaned as they all possessed some kind of string for unknown quantities/unavailable information. It was determined that we could not remove all string values as this is most likely a significant amount of our data. Some, like the date of the collision, were deemed too important not to have and were removed entirely. Others, like the road surface, were considered viable to train without. Subsequently, they were set to NaN. Post-processing revealed this was only 2% of the data.



P_ID

Code	Description	
01 - 99	01 - 99	
NN	Data element is not applicable	e.g. "dummy" person record created for parked cars
UU	Unknown	e.g. applies to runaway cars

Figure 1: Example of data field with string values. P_ID stands for passenger ID.

The labels were split into two fields: injury count and fatality count per collision. The distribution of these counts was represented by two histograms with a logarithmic y-axis. Both counts displayed an extreme positive skew, suggesting that the data is not as evenly distributed as we hoped. Some outliers were also present in the data, but these were removed to improve the accuracy of the model.

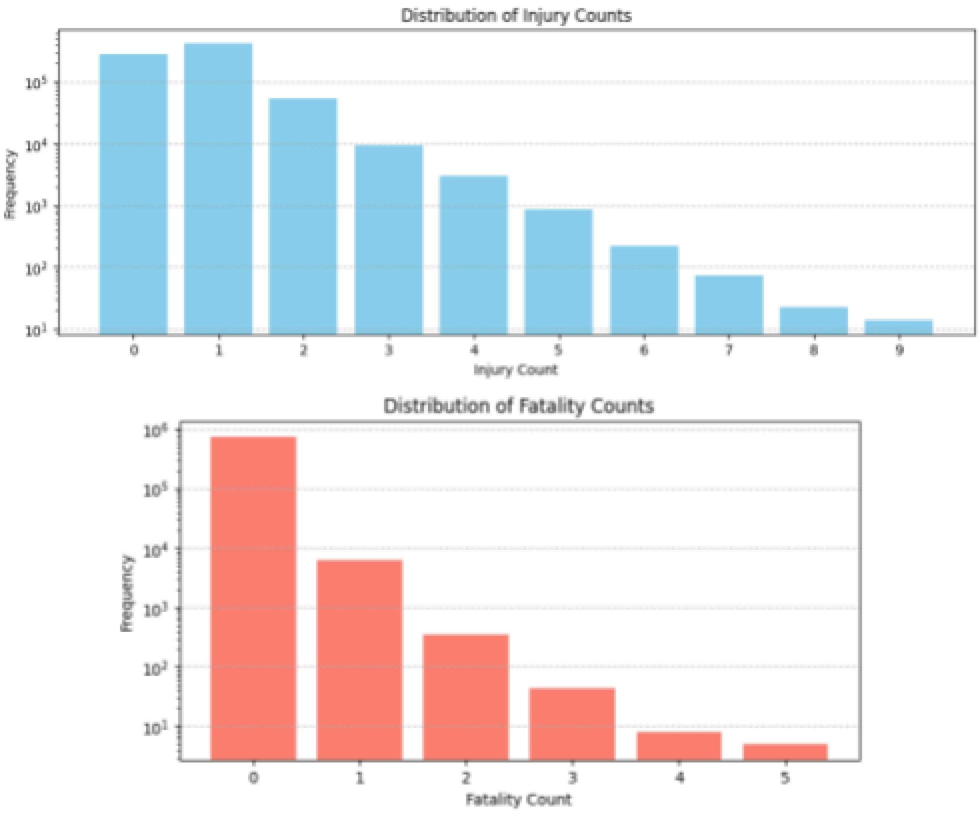


Figure 2: Histograms displaying data distribution.

MODEL

The main architecture is a gradient-boosting algorithm utilizing XGBoost. This algorithm builds a "forest" of decision tree models, starting with a base model. This base model is evaluated, then training samples are resampled and resampled based on this evaluation and fed into a new model (see f_1 in Figure 3 below). This is done continuously until n trees are trained, each tree better than the last. All of these models then collectively make predictions on the data.

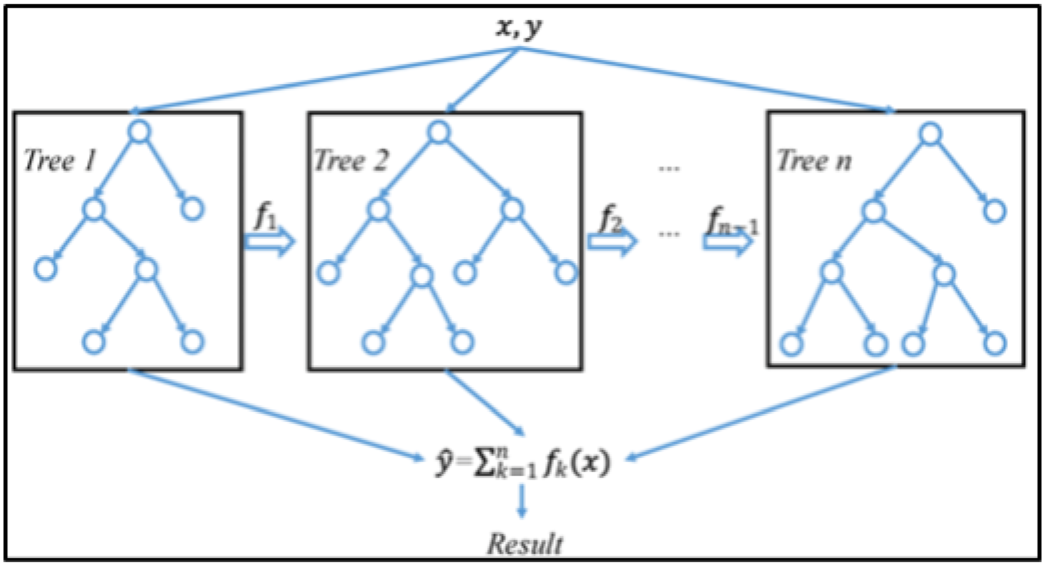


Figure 3: Illustration of the gradient boosting process, showing the iterative states to generate *n* trees and predict a result. [6]

XGBoost does not have adequate native support for multi-output modelling, so we had two options to regress to two separate outputs. The first method was to use sklearn's MultiOutputRegressor wrapper, which, when used with XGBoost's sklearn wrapper, can allow the model to perform multi-output regression. The reason this doesn't work as effectively is that the prediction of injuries and fatalities measures different levels of severity, so when the hyperparameters of this wrapped model are optimized, it settled for a set of hyperparameters that was in the middle of what each output would have needed. Therefore, a second method was chosen: use two models, one for injuries and one for fatalities.

Hyperparameter optimization was performed using sklearn's RandomizedSearchCV cross-validation method. For each hyperparameter to optimize, a range of values is provided, and this cross-validator selects random pairings of these ranges to determine the most optimal set of values. We chose to optimize n_estimators, max_depth, and learning_rate with this model. Fine-tuning these hyperparameters limits the amount that a model can overfit the data by limiting the number of gradient boosting rounds, reducing the depth of each tree, and reducing the rate with which successive decision trees learn from previous tree iterations, respectively.

EVALUATION

RMSE:

RMSE or Root Mean Squared Error measures the distance between predicted and actual values. Logically, a perfect model would have a RMSE of zero.

R2 Score:

The R2 score is the complement of RMSE and is also known as the coefficient of determination. It measures how well the model represents variability in the data, ranging from negative values (worse than taking the mean of all the data) to one (a perfect fit).

Results:

The results from the models can be seen in the table below. For both models, the R2 score performed poorly, indicating that they only predict a marginal percentage of the variance in the data. The RMSE for the injury model is less than the mean of the test data, indicating it is better than the mean for predicting results. However, the fatality model is two orders of magnitude higher than the mean, indicating that it is an incredibly poor predictor.

	RMSE	**Mean of test data**	**R2 Score**
Injury Model	0.5800	0.7454	0.2923
Fatality Model	0.1006	0.0094	0.0528

DISCUSSION

The beginning stages of the project were fairly smooth, including dataset selection, data cleansing and data validation. Significant challenges were encountered during the training and testing process. After selecting Gradient Boosting as the approach for this task, we chose XGBoost for its implementation. The first approach involved multi-output regression to handle the fatalities and injuries in a single model. However, the challenges with this approach are that the distribution of fatalities and injuries are very different, so the optimized hyperparameters do not fit either output well, leading to a less desirable R2 score and RMSE. To solve this problem, from then on, we chose to have two separate models, one for fatalities and one for injuries. The following approach was experimenting with different boosters from XGBoost, gbtree, gblinear, and dart. The booster with the best performance was gbtree, while gblinear and dart performed worse, especially for injuries and took too much time to train.

The following approach was to modify the loss function used by XGBoost for both models. The default loss function optimizes for Mean Squared Error (MSE) and has provided us with the best evaluation thus far. We changed the loss function to Mean Squared Logarithmic Error (MSLE) and found that this increased RMSE, indicating a worse performance. Due to this decrease in performance, we elected to stick with MSE as the loss function. Next, we adjusted the scale_pos_weight parameter to account for imbalances. This parameter is responsible for punishing incorrect values on “minority classes,” in regression, ranges that occur less, i.e. > 0. After changing scale_pos_weight to 10 instead of the default value of 1, the R2 score became slightly negative due to imbalances being overpunished. Therefore, we reverted the parameter to its default value of 1. Lastly, we experimented with removing outlier values, which provided minimal improvements to the scores as the large outlier values did not appear to impact the data substantially.

REFERENCES

[1] T. Canada, “Canadian Motor Vehicle Traffic Collision Statistics: 2022,” *Transport Canada*, May 02, 2024. https://tc.canada.ca/en/road-transportation/statistics-data/canadian-motor-vehicle-traffic-collision-statistics-2022

[2] A. Iranitalab and A. Khattak, “Comparison of four statistical and machine learning methods for crash severity prediction,” Accident Analysis &amp; Prevention, vol. 108, pp. 27–36, Nov. 2017. doi:10.1016/j.aap.2017.08.008

[3] T. Usman, L. Fu, and L. F. Miranda-Moreno, “Injury severity analysis: Comparison of multilevel logistic regression models and effects of collision data aggregation,” Journal of Modern Transportation, vol. 24, no. 1, pp. 73–87, Feb. 2016. doi:10.1007/s40534-016-0096-4

[4] V. Prasad, P. Vellivel, and R. Chandru, “Machine learning algorithms to model crash severity and collision(s),” 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), pp. 1–5, Nov. 2022. doi:10.1109/nkcon56289.2022.10126528

[5] S. Koley, S. Mondal, and P. Ghosal, “Smart prediction of severity in vehicular crashes: A machine learning approach,” 2022 5th International Conference on Computational Intelligence and Networks (CINE), pp. 1–5, Dec. 2022. doi:10.1109/cine56307.2022.10037474

[6] “XGBoost: A powerful machine learning framework,” CloudThat Resources, https://www.cloudthat.com/resources/blog/xgboost-a-powerful-machine-learning-framework (accessed Nov. 28, 2024).