Fatality Risk Prediction in Toronto Motor Vehicle Collisions using Gradient Boosting Decision Trees [DRAFT]

Abtin Kit Shirvani

February 2026

1 Introduction

Goal: accurate risk estimates and some insight into potential risk factors

2 Methodology

2.1 Data Sources

The KSI (Killed or Seriously Injured) Collisions dataset was sourced from the Toronto Police Service Public Safety Data Portal and contains all reported Toronto Motor Vehicle Collisions (MVCs) since 2006 where a person sustained a major or fatal injury.[1] A major injury is defined as a non-fatal injury that is severe enough to require the injured person to be admitted to the hospital, even if only for observation at the time of the collision (i.e. fractures, internal injury, severe cuts, crushing, burns, concussion, severe general shocks). A fatal injury is defined as sustaining bodily injuries resulting in death, where death occurs in less than 366 days as result of the collision.[*] The records contain information of the following:

- Geographic location[†]
- Environmental conditions
- Ages involved
- Types of vehicles involved
- Vehicle actions at the time of the crash
- Injury Severity[‡]
- Accident classification (i.e. `Fatal`, `Non-Fatal`)

Complementary geospatial datasets were incorporated for imputation and feature engineering:

[*] Does not include death from natural causes (e.g. heart attack, stroke, epilated seizure, etc.) or suicide.
[†] Exact locations of the crashes were offset to the nearest intersection for privacy concerns.
[‡] Only used for imputation to avoid leakage.

- **Road Infrastructure Data**[2]: Street-level spatial data for road classification sourced from Statistics Canada's road network repositories.
- **Health Services Provider Locations**[3]: Hospital spatial data obtained from the Ontario Ministry of Health's publicly available GeoHub.

All public datasets in this study were used in compliance with the Open Government Licence – Ontario v1.1 and Open Government Licence – Canada v2.0, which permit free use, modification, and distribution with appropriate attribution.[4;5]

2.2 Data preparation

Each observation in the KSI Collisions dataset corresponds to each person involved in the collision, regardless of the level of injury. Raw collision records were aggregated into unique events by combining identical entries based on three key identifiers: (1) the accident number, (2) the recorded date-time, and (3) geospatial coordinates (latitude and longitude). To preserve information during the aggregation, features that varied among individuals in the collision (i.e. Involved ages, Driver action, Manoeuver) were transformed from long to wide format. Collisions that occurred outside the City of Toronto, as indicated by `NSA`[§] neighbourhoods, were omitted. Missing feature values were imputed, where applicable, using the road infrastructure data or the nearest collision that contained the relevant data (spatial or temporal proximity depending on the feature). Otherwise, features with minimal missing entries were assigned to a pre-existing `Unknown` class. Features that were missing a large portion of data and/or were too difficult to extract any meaningful information were removed altogether. To reduce dimensionality and prevent unstable parameter estimates, rare categories were consolidated into an `Other` class. Missing accident class labels were imputed using the highest injury severity in each collision; events with ≥ 1 fatality were labelled `Fatal`, while those with only non-fatal injuries (given no missing injury data) were labelled `Non-Fatal`. The sparse `Property Damage O` accident class was consolidated with `Non-Fatal` to keep the classification binary, as these accidents still contained major injuries despite their label.[¶] Temporal features (i.e. Month, Day of week, Hour) were encoded using sine and cosine transformations to preserve their cyclical nature. To assess the impact of proximity to emergency care, the distance to the nearest ER along with the associated hospital name were added as features. Distance was computed using Manhattan (L^1) geometry to more closely resemble grid-like city driving. The NAD83 CSRS[‖] was used throughout the analysis to produce more accurate spatial joins and distance calculations. The final dataset was partitioned into 80% training, 10% calibration, and 10% test splits, using stratified sampling.

2.3 Model Training

2.3.1 Algorithm Selection

Gradient Boosting Decision Trees (GBDTs) were chosen for their strong predictive performance on structured data and their demonstrated advantages over classical statistical methods and other ML/DL[**] approaches in binary classification tasks.[6–9] GBDTs are particularly effective when complex, non-linear relationships exist and sufficient training data is available, however what is consid-

[§] Not Specified Area.

[¶] Inferred to mean `Property Damage Only` due to ambiguous documentation.

[‖] Canadian Spatial Reference System (EPSG:2958).

[**] Machine Learning/Deep Learning

ered *sufficient* varies across domains. Benchmark studies and applications in related fields suggest that GBDTs typically outperform classical methods when sample sizes reach the order of thousands to tens of thousands.[10–12] While no widely cited traffic safety study defines a universal sample-size threshold, several have reported strong predictive performance with as few as 1,200 observations, supporting the applicability of GBDTs to moderate-sized collision datasets.[13;14] Although state-of-the-art neural networks can achieve comparable performance on tabular data, GBDTs have been shown to better handle skewed feature distributions and other dataset irregularities.[15] Moreover, GBDTs (or more generally DTs) are robust to monotonic feature transformations, eliminating the need for scaling or normalization.[16]

2.3.2 Hyperparameter Tuning

Hyperparameter tuning was performed using 10 rounds of Optuna's Bayesian TPE[††] sampling algorithm to balance exploration of the parameter space and computational cost. The search space included key GBDT parameters: tree depth, learning rate, L_2 leaf regularization, random subspace method (RSM), and class weights. Early stopping was set to 100 rounds to prevent overfitting, and all tuning was conducted within the cross-validation framework to ensure unbiased performance estimates.

2.3.3 Cross-Validation

Model performance was evaluated using 5-fold stratified cross-validation with a fixed seed (42). Class weights were applied to mitigate bias towards the majority (non-fatal) class. Several custom loss functions designed for imbalanced datasets, along with log loss, were evaluated using Precision-Recall Area Under the Curve (PR-AUC) as the primary performance metric. PR-AUC was favoured over other common evaluation metrics (e.g. Log-loss, F_1, AUC) for its threshold independence and robustness against class imbalance.[17;18] Decision threshold optimization was omitted in the main analysis, as the primary objective of the model was risk prediction (see Appendix B.1).

2.4 Calibration

GBDTs often produce poorly calibrated probability estimates, but applying post-hoc calibration can substantially improve their reliability.[19] Prior studies have shown that GBDTs can achieve lower Brier scores and expected calibration errors (ECE) than both generalized linear models and deep learning approaches in clinical and credit risk applications.[10;20]

Common calibration methods include Platt scaling, isotonic regression, beta calibration, and temperature scaling, with relative performances varying across classifiers and datasets.[21] In this study, Platt scaling (also known as sigmoid or logistic calibration) was selected due to its low computational cost and robust performance with limited calibration data.[22–24] Model outputs on the validation set were verified to approximately satisfy the logistic assumption (See Appendix B). Platt scaling was implemented using the `netcal` package. To prevent data leakage, the calibration model was trained on a separate calibration set. Performance was evaluated visually using calibration plots and quantitatively using ECE and Brier score.

[††] Tree-structured Parzen Estimator

2.5 Loss functions

2.5.1 Log loss

Logarithmic loss (also known as Cross-Entropy or Log loss) is a widely used loss function for classification tasks. It quantifies the discrepancy between predicted class probabilities and true labels, imposing higher penalties for greater deviations. However, a significant limitation of Log loss is its sensitivity to class imbalance, as misclassification errors from the majority class tend to dominate the loss function, overshadowing errors in the minority class.[25] To mitigate this issue, a common approach involves applying class-weighted adjustments, where a multiplicative weight is assigned to the minority class to increase its influence during model training.[26;27] The weighted Log loss function for binary classification is defined as the following:

$$\mathcal{L}_{Log}(p_i, y_i; \boldsymbol{\alpha}) = -\left[\alpha_1 \cdot y_i \log(p_i) + \alpha_0 \cdot (1 - y_i)\log(1 - p_i)\right] \quad \forall\, i = 1, \ldots, n$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_1]^T \in \mathbb{R}^2_{>0}$ is a tunable weight term, $p_i \in (0, 1)$ is the i^{th} predicted probability for class 1, and $y_i \in \{0, 1\}$ is the i^{th} true label.

2.5.2 Focal loss

Focal loss (FL) is an extension of Log loss designed to address class imbalance by down-weighting the contribution of easily classified observations.[28] The loss reduces the influence of high-confidence predictions (where $p_i \approx y_i$), forcing the model to prioritize misclassified or uncertain samples. The Focal loss for binary classification with class weights is defined as the following:

$$\mathcal{L}_{Focal}(p_i, y_i; \boldsymbol{\alpha}, \gamma) = -\left[\alpha_1 \cdot (1 - p_i)^\gamma \cdot y_i \log(p_i) + \alpha_0 \cdot p_i^\gamma \cdot (1 - y_i)\log(1 - p_i)\right] \quad \forall\, i = 1, \ldots, n$$

where $\gamma \geq 0$ is a hyperparameter that controls the down-weighting, i.e., as γ increases, the contribution of well-classified samples to the overall loss approaches 0.

2.5.3 Logit-Adjusted loss

Instead of using weights to adjust class influence, Logit-Adjusted (LA) loss directly modifies the raw logits (log-odds) based on a priori class probabilities.[29] This effectively places a penalty on the majority class logit, and a boost on the minority class logit, counteracting bias towards the majority class. The LA loss for binary classification with class weights is defined as the following:

$$\mathcal{L}_{LA}(z_i, y_i; \boldsymbol{\alpha}, \boldsymbol{\pi}, \tau) = -\left[\alpha_1 \cdot y_i \log\left(\sigma(z_i + \tau \log\frac{\pi_1}{\pi_0})\right) + \alpha_0 \cdot (1 - y_i)\log\left(1 - \sigma(z_i + \tau \log\frac{\pi_1}{\pi_0})\right)\right] \quad \forall\, i = 1, \ldots, n$$

where $\sigma(\cdot)$ is the sigmoid function[‡‡], z_i is the raw logit for the i^{th} sample, $\boldsymbol{\pi} = [\pi_0, \pi_1]^T$ is the a priori probabilities for majority and minority classes respectively, and $\tau \geq 0$ is a hyperparameter that controls the scaling of $\boldsymbol{\pi}$. Setting $\tau = 1$ comes with the added theoretical guarantee that the model converges towards the Bayes-optimal classifier for balanced error.[29]

[‡‡] The LA paper uses softmax, but in the binary case, it can be simplified to a sigmoid

2.5.4 Label-Distribution-Aware-Margin loss

Similar to LA loss, Label-Distribution-Aware-Margin (LDAM) loss directly modifies the raw logits, but instead uses a class-dependent offset term.[30] This also effectively shifts the decision boundary towards the majority class,

$$\mathcal{L}_{LDAM}(z_i, y_i; \boldsymbol{\alpha}, C, \mathbf{n}) = -\left[\alpha_1 \cdot y_i \log\left(\sigma(z_i - \frac{C}{\sqrt[4]{n_1}})\right) + \alpha_0 \cdot (1-y_i)\log\left(1 - \sigma(z_i + \frac{C}{\sqrt[4]{n_0}})\right)\right] \quad \forall\, i = 1, \ldots, n$$

2.6 GBDT Theory

The main idea of boosting is to iteratively build an ensemble of weak models[§§], with each new model targeting the errors of the previous one. Given enough iterations, this correction process can lead to a surprisingly strong model. GBDTs are a specific type of boosting where the models being fit are decision trees, and the errors from the previous models are calculated using the gradient of a loss function. Several widely-used GBDT implementations have emerged in recent years, including XGBoost, LightGBM, and CatBoost.[31–33] While these modern implementations introduce various optimizations and enhancements, their theoretical foundations trace back to Jerome Friedman's work on gradient boosting. The following in-depth summary of the process of training a GBDT was sourced from Friedman's 1999 papers titled *Greedy function approximation: A gradient boosting machine* and *Stochastic Gradient Boosting.*[34;35]

2.6.1 Initialization

We begin with a collection of i.i.d. training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where each $\mathbf{x}_i \in \mathcal{X}$ is a high-dimensional feature vector and $y_i \in \{0, 1\}$ is its corresponding label. The loss function (log-loss) that measures prediction error is given by:

$$L(\gamma; y_i) = \log(1 + e^{\gamma}) - y_i\gamma \quad \text{for all } i = 1, \ldots, n, \tag{1}$$

where γ represents the log-odds (logit) of the probability estimate (See Appendix A.1 for the complete derivation of (1)). Formally, we define:

$$F_m : \mathcal{X} \to \mathbb{R} \tag{2}$$

as the prediction model at the m^{th} iteration. The initialization of this model begins with a constant prediction value (log-odds) obtained by minimizing the total loss across all training samples:

$$F_0(\mathbf{x}) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(\gamma; y_i) \tag{3}$$

2.6.2 Iterations

To measure the error of the previous iteration, we compute n pseudo-residuals (1 for each training sample) using the negative gradient of the loss function from the previous iteration:

$$r_{im} = -\left[\frac{\partial L(F(\mathbf{x}_i); y_i)}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}) = F_{m-1}(\mathbf{x})}, \quad \text{for all } i = 1, ..., n \tag{4}$$

[§§] also referred to as weak learners

where $\frac{\partial L(F(\mathbf{x}_i);y_i)}{\partial F(\mathbf{x}_i)}$ is the loss function gradient, calculated using the log-odds estimates of the previous iteration. The intuition behind taking the negative gradient of the loss function being that we move towards the minimum of the loss function. These pseudo-residuals are then used to fit a regression tree (the current iteration's model). We denote the leaves (terminal regions) for the newly fitted regression tree as R_{jm} for $j = 1, ..., J_m$ where J_m is the total number of leaves, and compute a new log-odds estimate (γ_{jm}) for each leaf.

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{i:\mathbf{x}_i \in R_{jm}} L(F_{m-1}(\mathbf{x}_i) + \gamma; y_i) \tag{5}$$

The minimization problem in (5) can be solved using various numerical optimization techniques such as gradient descent, line search, or Newton's method. One simple yet effective approach used by XGBoost is the second-order Taylor approximation of the objective function.[31] This approach works well because (a) the learning rate ensures small enough step sizes to maintain the validity of the approximations and (b) the formulation leads to an elegant closed-form solution:

$$\gamma_{jm} = \frac{\sum_{i:\mathbf{x}_i \in R_{jm}} r_{im}}{\sum_{i:\mathbf{x}_i \in R_{jm}} p_i(1-p_i)} \tag{6}$$

where $p_i = \frac{e^{F_{m-1}(\mathbf{x}_i)}}{1+e^{F_{m-1}(\mathbf{x}_i)}}$ and denotes the probability of being class 1 for the i^{th} sample, based on the log-odds estimates of the previous iteration. The term in the numerator is the sum of the pseudo-residuals in leaf j (See Appendix A.2 for the complete derivation of (6)). The model at iteration m is updated additively using the log-odds estimates of the previous iteration.

$$F_m(\mathbf{x}_i) = F_{m-1}(\mathbf{x}_i) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}(\mathbf{x}_i \in R_{jm}), \quad \text{for all } i = 1, \ldots, n \tag{7}$$

where ν is the learning rate. The second term essentially sums up the pseudo-residuals that match the given $\mathbf{x}_i$, and its effect on the current iteration is shrunk by the learning rate. Apart from the initialization, this whole process repeats for M iterations. The final classification is determined using the last predicted probabilities ($p_i = \frac{e^{F_M(\mathbf{x}_i)}}{1+e^{F_M(\mathbf{x}_i)}}$) given $\mathbf{x}_i$, with some cutoff threshold (typically 0.5).

$$\hat{y}_i = \begin{cases} 1 & p_i \geq 0.5 \\ 0 & otherwise \end{cases} \tag{8}$$

Algorithm 1: Gradient Boosting Decision Trees

```
Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, and a differentiable Loss Function $L(F(\mathbf{x}); y_i)$
Output: $F_M(\mathbf{x})$
Initialize: $F_0(\mathbf{x}) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(\gamma; y_i)$
for $m = 1$ to $M$ do
    1. Compute pseudo-residuals:
       $r_{im} = -\left[\frac{\partial L(F(\mathbf{x}_i); y_i)}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}_i)=F_{m-1}(\mathbf{x}_i)}, \quad \forall\, i = 1, \ldots, n$
    2. Fit regression tree:
       Train weak learner using $\{(\mathbf{x}_i, r_{im})\}_{i=1}^n$, producing terminal regions $\{R_{jm}\}_{j=1}^{J_m}$
    3. Compute leaf weights:
       $\gamma_{jm} = \arg\min_{\gamma} \sum_{i:\mathbf{x}_i \in R_{jm}} L(F_{m-1}(\mathbf{x}_i) + \gamma; y_i), \quad \forall\, j = 1, \ldots, J_m$
    4. Update model:
    $F_m(\mathbf{x}_i) = F_{m-1}(\mathbf{x}_i) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}(\mathbf{x}_i \in R_{jm}), \quad \forall\, i = 1, \ldots, n$
end
```

3 Results

3.1 Model selection

Table 1 summarizes cross-validation performance across algorithms and loss functions on the training set using PR-AUC as the evaluation metric. Among the candidate models, CatBoost trained with LDAM loss achieved the highest mean PR-AUC (0.312 ± 0.017).

Table 1: Summary of Cross Validation Results across algorithms and loss functions using PR-AUC evaluation metric

Algorithm	**Log loss**	**LDAM**	**Focal loss**	**LA**
CatBoost	0.304 ± 0.023	$\mathbf{0.312 \pm 0.017}$*	0.286 ± 0.021	0.261 ± 0.024
XGBoost	0.257 ± 0.010	0.243 ± 0.013	0.265 ± 0.008	0.276 ± 0.008
LightGBM	0.253 ± 0.030	0.266 ± 0.026	0.235 ± 0.017	0.234 ± 0.019

**Highest mean PR-AUC*

Based on overall discrimination and stability across folds, CatBoost with LDAM loss was selected as the final model. When evaluated on the held-out test set, the model achieved a PR-AUC of 0.341 (Figure 1), representing a 0.202 improvement over the baseline performance of a naive classifier (0.139).

3.2 Calibration

Calibration performance of the final model is shown in Figure 2. Predicted probabilities were grouped into deciles of predicted risk, and observed fatality rates were computed within each bin.

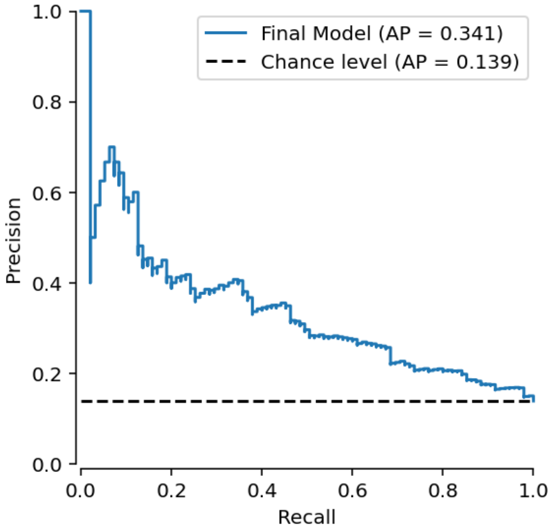


Figure 1: Precision-Recall plot

Due to low prevalence of fatalities, the axes were restricted to the range of predicted probabilities observed in the test set (< 0.4). A side-by-side comparison with the uncalibrated model is provided in Appendix B. The curve is generally monotonically increasing, apart from a few oscillations in the low-to-mid risk range. The mid-to-high risk range exhibits mild underestimation of fatality risk.

Table 2 reports discrimination and calibration metrics for the final model on the test set, before and after calibration. PR-AUC is reported only for the uncalibrated model, as it is invariant to calibration and depends solely on the relative ranking of predictions. Calibration substantially improved probabilistic accuracy, reducing the Brier score from 0.180 to 0.108 ($\Delta = 0.072$) and the expected calibration error (ECE) from 0.255 to 0.007 ($\Delta = 0.248$).

Table 2: Final calibrated model metrics on the test set

Metric	**Value**	
	Uncalibrated	**Calibrated**
PR-AUC	0.341	-
Brier score	0.180	0.108
ECE	0.255	0.007

3.3 Feature Importance

Figure 3 displays SHAP-based feature importance results for the final model. The bar plot (left) ranks the top 12 features by mean absolute SHAP value, while the beeswarm plot (right) illustrates the distribution and direction of feature effects on model predictions. Impact type was the most influential predictor, followed by calendar year, district, and the number of persons involved in the

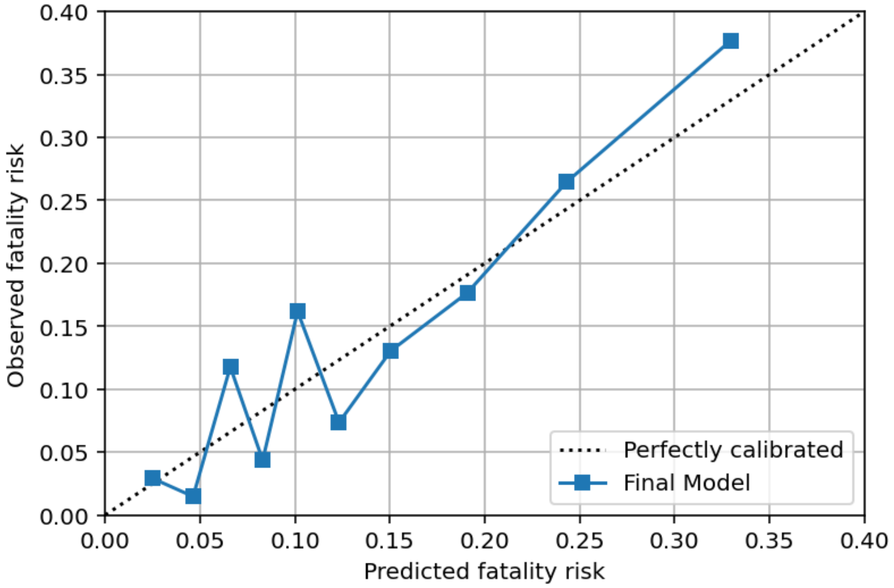


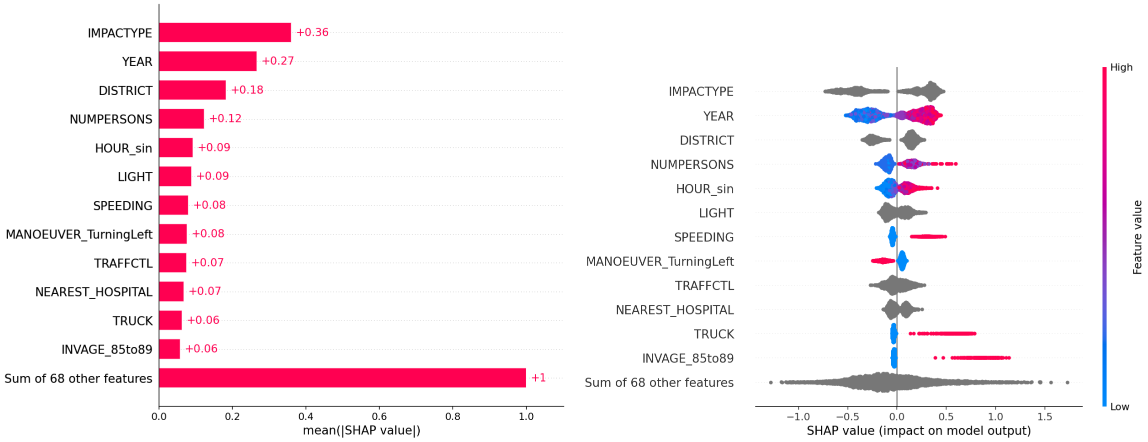
Figure 2: Calibration plot on test set

accident. Higher values of `YEAR` and `NUMPERSONS` were generally associated with increased predicted risk. Positive values of the sinusoidal hour-of-day encoding (`HOUR_sin`) corresponded to higher risk, indicating that accidents occurring earlier in the day (AM) tended to have elevated predicted risk. Speeding, truck involvement, and the presence of at least one individual aged 85–89 years were also linked to higher predicted risk. In contrast, collisions involving at least one motor vehicle executing a left turn were among the few scenarios associated with lower predicted risk. Note that the observed associations reflect patterns learned by the model and should not be interpreted as causal effects.

4 Discussion

The final model demonstrated good generalization to unseen data, indicating that the selected features captured a substantial portion of the predictive signal relevant to fatal collision risk. The similarity in cross validation scores across algorithms and loss functions indicates that further performance gains are more likely to arise from additional data or feature engineering rather than changes in model architecture. Minor performance differences across models may also be attributable to the non-exhaustive nature of the hyperparameter search. CatBoost models had a higher tendency to overfit, likely due to the algorithm's method of data-partitioning during training and the low prevalence of fatal outcomes.

Several potentially informative domains were not explicitly represented in the feature set. Temporal features served primarily as indirect proxies for latent factors such as traffic volume and congestion. Incorporating complementary data sources that cover these risk factors directly may improve predictive performance. Similarly, although visibility and light features broadly reflected



(a) Mean absolute SHAP values

(b) Beeswarm plot

Figure 3: SHAP Feature Importances for Final Model

weather conditions, the model may benefit from high-resolution meteorological data, including precise measures of precipitation and temperature rather than relying on qualitative reports.

Post-hoc calibration substantially improved the reliability of predicted probabilities. Overestimation in the low-to-mid risk ranges may lead to conservative risk assessments, whereas mid-to-high risk scenarios may underestimate true risk. Upper tail estimates (> 0.25) are likely to be less stable due to the rarity of fatal outcomes and the high variance of predicted probabilities within mid-to-high risk bins. Future studies may investigate the use of Bayesian calibration techniques such as Bayesian Binning into Quantiles (BBQ) to improve stability in high-risk regions.[36]

The analysis of feature importance using SHAP values identified several predictors consistently associated with elevated fatality risk, including speeding, trucks and elderly involved, calendar year, and the number of individuals involved in a collision. Although SHAP-based explanations derived from observational data should not be interpreted causally, these findings are broadly consistent with existing literature on traffic safety, supporting the plausibility of the model's learned associations.

The modelling framework assumes independence between collision events, which is unlikely to hold due to spatial and temporal clustering of crashes.[37;38] While such dependence primarily affects statistical inference in parametric models, ignoring clustering may also lead to overly optimistic risk estimates if correlated events appear in both training and test sets.[39] Future studies may benefit from explicitly modelling spatio-temporal dependence or adopting clustered validation strategies.

Finally, it is important to note that the estimated risks are conditional on a collision occurring. The marginal rate of fatal collisions in Toronto is only a few deaths per 100,000 people per year, which corresponds to a marginal risk that is orders of magnitude lower than the conditional risk.[40] Maintaining this contextual grounding is essential for proper interpretation and application of the results.

Post-hoc calibration does not improve robustness against out-of-distribution inputs.[41] Risk estimates only cover aleatoric (first-order) uncertainty. Future work can explore predicting uncertainty of risk estimates using bayesian methods and credal sets.

Supplementary Information

Availability of Data and Materials

The full data and code required to reproduce the results described in this study are available at the following GitHub repository: https://github.com/AShirvani01/CrashClass

Attribution Statement

Contains information licensed under the Open Government Licence – Canada and Open Government Licence – Ontario.

Competing Interests

The authors declare no conflicts of interest.

References

[1] Toronto Police Service. Killed and seriously injured, 2024. URL `https://data.torontopolice.on.ca/datasets/TorontoPS::killed-and-seriously-injured/about`.

[2] Statistics Canada. Road network files, 2024. URL `https://www12.statcan.gc.ca/census-recensement/2021/geo/sip-pis/rnf-frr/index-eng.cfm`.

[3] Ontario Ministry of Health. Ministry of health service provider locations, 2024. URL `https://geohub.lio.gov.on.ca/datasets/lio::ministry-of-health-service-provider-locations-1/about`.

[4] Open government license - ontario, 2025. URL `https://data.torontopolice.on.ca/pages/licence`.

[5] Open government license - canada, 2025. URL `https://open.canada.ca/en/open-government-licence-canada`.

[6] Assaf Shmuel, Oren Glickman, and Teddy Lazebnik. A comprehensive benchmark of machine and deep learning across diverse tabular datasets, 2024. URL `https://arxiv.org/abs/2408.14817`.

[7] Marc Schmitt. Deep learning vs. gradient boosting: Benchmarking state-of-the-art machine learning algorithms for credit scoring, 2022. URL `https://arxiv.org/abs/2205.10535`.

[8] Roel Henckaerts, Marie-Pier Côté, Katrien Antonio, and Roel Verbelen. Boosting insights in insurance tariff plans with tree-based machine learning methods. *N. Am. Actuar. J.*, 25(2): 255–285, April 2021. doi: https://doi.org/10.1080/10920277.2020.1745656.

[9] Getahun Mulugeta, Temesgen Zewotir, Awoke Seyoum Tegegne, Leja Hamza Juhar, and Mahteme Bekele Muleta. Classification of imbalanced data using machine learning algorithms to predict the risk of renal graft failures in ethiopia. *BMC Med. Inform. Decis. Mak.*, 23(1): 98, May 2023. doi: https://doi.org/10.1186/s12911-023-02185-5.

[10] Abdulazeez Alabi, Olajide Akinpeloye, Osayimwense Izinyon, Tope Amusa, and Akinwale Famotire. From logistic regression to foundation models: Factors associated with improved forecasts. *Cureus*, 17(11):e96669, November 2025. doi: 10.7759/cureus.96669.

[11] Hiroe Seto, Asuka Oyama, Shuji Kitora, Hiroshi Toki, Ryohei Yamamoto, Jun'ichi Kotoku, Akihiro Haga, Maki Shinzawa, Miyae Yamakawa, Sakiko Fukui, and Toshiki Moriyama. Gradient boosting decision tree becomes more reliable than logistic regression in predicting probability for diabetes with big data. *Sci. Rep.*, 12(1):15889, October 2022. doi: https://doi.org/10.1038/s41598-022-20149-z.

[12] Scott Silvey and Jinze Liu. Sample size requirements for popular classification algorithms in tabular clinical data: Empirical study. *Journal of Medical Internet Research*, 26:e60231, 2024. doi: 10.2196/60231. URL `https://www.jmir.org/2024/1/e60231/`.

[13] Zhe Zhang, Wentao Wu, Qi Cao, Jianhua Song, Jingfeng Ma, Gang Ren, and Changjian Wu. Deciphering the crash mechanisms in autonomous vehicle systems via explainable ai. *Systems*, 14(1), 2026. ISSN 2079-8954. doi: 10.3390/systems14010104. URL `https://www.mdpi.com/2079-8954/14/1/104`.

[14] Abraham Keffale Mengistu, Andualem Enyew Gedefaw, Nebebe Demis Baykemagn, Agmasie Damtew Walle, and Tirualem Zeleke Yehuala. Predicting car accident severity in northwest ethiopia: a machine learning approach leveraging driver, environmental, and road conditions. *Scientific Reports*, 15:21913, 2025. doi: 10.1038/s41598-025-08005-2. URL `https://doi.org/10.1038/s41598-025-08005-2`.

[15] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Benjamin Feuer, Chinmay Hegde, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data?, 2024. URL `https://arxiv.org/abs/2305.02997`.

[16] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370.

[17] Eve Richardson, Raphael Trevizani, Jason A. Greenbaum, Hannah Carter, Morten Nielsen, and Bjoern Peters. The receiver operating characteristic curve accurately assesses imbalanced datasets. *Patterns*, 5(6):100994, 2024. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2024.100994. URL `https://www.sciencedirect.com/science/article/pii/S2666389924001090`.

[18] Helen R. Sofaer, Jennifer A. Hoeting, and Catherine S. Jarnevich. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4):565–577, 2019. doi: https://doi.org/10.1111/2041-210X.13140. URL `https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13140`.

[19] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 161–168, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143865. URL `https://doi.org/10.1145/1143844.1143865`.

[20] Pedro G. Fonseca and Hugo D. Lopes. Calibration of machine learning classifiers for probability of default modelling, 2017. URL `https://arxiv.org/abs/1710.08901`.

[21] Valery Manokhin and Daniel Grønhaug. Classifier calibration at scale: An empirical study of model-agnostic post-hoc methods, 2026. URL `https://arxiv.org/abs/2601.19944`.

[22] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. URL `https://arxiv.org/abs/1706.04599`.

[23] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 623–631. PMLR, 20–22 Apr 2017. URL `https://proceedings.mlr.press/v54/kull17a.html`.

[24] Björn Böken. On the appropriateness of platt scaling in classifier calibration. *Inf. Syst.*, 95 (101641):101641, January 2021. doi: https://doi.org/10.1016/j.is.2020.101641.

[25] YANMIN SUN, ANDREW K. C. WONG, and MOHAMED S. KAMEL. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009. doi: 10.1142/S0218001409007326. URL `https://doi.org/10.1142/S0218001409007326`.

[26] Mohammad Reza Rezaei-Dastjerdehei, Amirmohammad Mijani, and Emad Fatemizadeh. Addressing imbalance in multi-label classification using weighted cross entropy loss function. In *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, pages 333–338, 2020. doi: 10.1109/ICBME51989.2020.9319440.

[27] Yuri Sousa Aurelio, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. Learning from imbalanced data sets with weighted cross-entropy function. *Neural Process. Lett.*, 50(2):1937–1949, October 2019. doi: https://doi.org/10.1007/s11063-018-09977-1.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. URL `https://arxiv.org/abs/1708.02002`.

[29] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment, 2021. URL `https://arxiv.org/abs/2007.07314`.

[30] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *CoRR*, abs/1906.07413, 2019. URL `http://arxiv.org/abs/1906.07413`.

[31] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL `http://arxiv.org/abs/1603.02754`.

[32] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[33] Anna Veronika Dorogush, Andrey Gulin, Gleb Gusev, Nikita Kazeev, Liudmila Ostroumova Prokhorenkova, and Aleksandr Vorobev. Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516, 2017. URL `http://arxiv.org/abs/1706.09516`.

[34] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001. doi: 10.1214/aos/1013203451. URL `https://doi.org/10.1214/aos/1013203451`.

[35] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics Data Analysis*, 38:367–378, 02 2002. doi: 10.1016/S0167-9473(01)00065-2.

[36] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proc. Conf. AAAI Artif. Intell.*, 2015:2901–2907, January 2015.

[37] Wondwossen Taddesse Gedamu, Uwe Plank-Wiedenbeck, and Bikila Teklu Wodajo. Spatio-temporal analysis of road traffic crashes by severity. *Transportation Engineering*, 20:100327, 2025. ISSN 2666-691X. doi: https://doi.org/10.1016/j.treng.2025.100327. URL https://www.sciencedirect.com/science/article/pii/S2666691X25000272.

[38] Wondwossen Taddesse Gedamu, Uwe Plank-Wiedenbeck, and Bikila Teklu Wodajo. A spatial autocorrelation analysis of road traffic crash by severity using moran's i spatial statistics: A comparative study of addis ababa and berlin cities. *Accident Analysis Prevention*, 200: 107535, 2024. ISSN 0001-4575. doi: https://doi.org/10.1016/j.aap.2024.107535. URL https://www.sciencedirect.com/science/article/pii/S0001457524000800.

[39] Mariana Oliveira, Luís Torgo, and Vítor Santos Costa. Evaluation procedures for forecasting with spatiotemporal data. *Mathematics*, 9(6), 2021. ISSN 2227-7390. doi: 10.3390/math9060691. URL https://www.mdpi.com/2227-7390/9/6/691.

[40] City of Toronto Transportation Services. Fatalities – vision zero. https://www.toronto.ca/services-payments/streets-parking-transportation/road-safety/vision-zero/vision-zero-dashboard/fatalities-vision-zero/, 2026.

[41] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019. URL https://arxiv.org/abs/1906.02530.

[42] Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *IEEE Access*, 9:78368–78381, 2021. doi: 10.1109/ACCESS.2021.3084050.

A Derivations

This section provides mathematical details underlying the methods referenced in the main text.

A.1 Loss function

The log-loss function is derived from the binomial likelihood:

$$\ell(p; y_1, \ldots, y_n) = \prod_{i=1}^{n} [p^{y_i} + (1-p)^{1-y_i}]$$

Applying a log transformation yields the log-likelihood:

$$\log[\ell(p; y_1, ..., y_n)] = \sum_{i=1}^{n} [y_i \log(p) + (1-y_i)\log(1-p)]$$

The per-sample loss can be expressed as:

$$\log[\ell(p; y_i)] = y_i \log(\frac{p}{1-p}) - \log(1-p) \quad \forall\ i = 1, \ldots, n$$

Expressing the log-likelihood in terms of the log-odds $\gamma = \log(\frac{p}{1-p})$ and taking the negative converts it to a minimization problem, yielding the final loss function:

$$\begin{aligned} -\log[\ell(\gamma; y_i)] &= \log(1 + e^{\gamma}) - y_i\gamma \quad \forall\ i = 1, \ldots, n \\ &= L(\gamma; y_i) \end{aligned}$$

From this derivation, we can see that minimizing the log-loss is analogous to maximizing the binomial likelihood.

A.2 Log-odds optimization

The 2nd-order taylor polynomial approximation of $L(F_{m-1}(\mathbf{x}_i) + \gamma; y_i)$ is

$$L(F_{m-1}(\mathbf{x}_i); y_i) + \gamma \nabla_F L(F_{m-1}(\mathbf{x}_i); y_i) + \frac{1}{2}\gamma^2 \nabla_F^2 L(F_{m-1}(\mathbf{x}_i); y_i)$$

where

$$L(F_{m-1}(\mathbf{x}_i); y_i) = \log(1 + e^{F_{m-1}(\mathbf{x}_i)}) - y_i F_{m-1}(\mathbf{x}_i)$$

The first and second derivatives are:

$$\begin{aligned} \nabla_F L(F_{m-1}(\mathbf{x}_i); y_i) &= \nabla_F \left[\log(1 + e^{F_{m-1}(\mathbf{x}_i)}) - y_i F_{m-1}(\mathbf{x}_i)\right] \\ &= \frac{e^{F_{m-1}(\mathbf{x}_i)}}{1 + e^{F_{m-1}(\mathbf{x}_i)}} - y_i \\ \nabla_F^2 L(F_{m-1}(\mathbf{x}_i); y_i) &= \nabla_F \left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{1 + e^{F_{m-1}(\mathbf{x}_i)}} - y_i\right] \\ &= \frac{e^{F_{m-1}(\mathbf{x}_i)}}{(1 + e^{F_{m-1}(\mathbf{x}_i)})^2} \end{aligned}$$

Substituting these into the approximation:

$$L(F_{m-1}(\mathbf{x}_i)+\gamma;y_i) \approx \log(1+e^{F_{m-1}(\mathbf{x}_i)}) - y_iF_{m-1}(\mathbf{x}_i) + \gamma\left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{1+e^{F_{m-1}(\mathbf{x}_i)}} - y_i\right] + \frac{1}{2}\gamma^2\left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{(1+e^{F_{m-1}(\mathbf{x}_i)})^2}\right]$$

To find the optimal γ_{jm} that minimizes the loss over the region R_{jm}, we solve:

$$\begin{aligned}
\gamma_{jm} &= \underset{\gamma}{\operatorname{argmin}} \sum_{i:\mathbf{x}_i \in R_{jm}} L(F_{m-1}(\mathbf{x}_i)+\gamma;y_i) \\
&= \nabla_\gamma\left[\sum_{i:\mathbf{x}_i \in R_{jm}} L(F_{m-1}(\mathbf{x}_i)+\gamma;y_i)\right] \\
&= \sum_{i:\mathbf{x}_i \in R_{jm}} \nabla_\gamma L(F_{m-1}(\mathbf{x}_i)+\gamma;y_i)
\end{aligned}$$

Using the Taylor approximation and setting the gradient to zero:

$$\begin{aligned}
\gamma_{jm} &\approx \sum_{i:\mathbf{x}_i \in R_{jm}} \nabla_\gamma\left[\log(1+e^{F_{m-1}(\mathbf{x}_i)}) - y_iF_{m-1}(\mathbf{x}_i) + \gamma\left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{1+e^{F_{m-1}(\mathbf{x}_i)}} - y_i\right] + \frac{1}{2}\gamma^2\left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{(1+e^{F_{m-1}(\mathbf{x}_i)})^2}\right]\right] \\
0 &= \sum_{i:\mathbf{x}_i \in R_{jm}} \left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{1+e^{F_{m-1}(\mathbf{x}_i)}} - y_i + \hat{\gamma}\left[\frac{e^{F_{m-1}(\mathbf{x}_i)}}{(1+e^{F_{m-1}(\mathbf{x}_i)})^2}\right]\right] \\
0 &= \sum_{i:\mathbf{x}_i \in R_{jm}} p_i - \sum_{i:\mathbf{x}_i \in R_{jm}} y_i + \hat{\gamma} \sum_{i:\mathbf{x}_i \in R_{jm}} p_i(1-p_i)
\end{aligned}$$

where $p_i = \frac{e^{F_{m-1}(\mathbf{x}_i)}}{1+e^{F_{m-1}(\mathbf{x}_i)}}$ and $1-p_i = \frac{1}{1+e^{F_{m-1}(\mathbf{x}_i)}}$. Solving for $\hat{\gamma}$:

$$\hat{\gamma}_{jm} = \frac{\sum_{i:\mathbf{x}_i \in R_{jm}} [p_i - y_i]}{\sum_{i:\mathbf{x}_i \in R_{jm}} p_i(1-p_i)}$$

A.3 LA loss

LA pairwise margin loss as defined in the paper[29]:

$$\mathcal{L}_{LA}(y, f(x)) = \alpha_y \cdot \log\left[1 + \sum_{y' \neq y} e^{\tau \Delta_{yy'} + f_{y'}(x) - f_y(x)}\right] \qquad \Delta_{yy'} = \log\left(\frac{\pi_{y'}}{\pi_y}\right)$$

$$\begin{aligned}
\mathcal{L}_{LA}(1, f(x)) &= \alpha_1 \cdot \log\left[1 + e^{\tau \Delta_{10} + f_0(x) - f_1(x)}\right] \\
&= \alpha_1 \cdot \log\left[1 + e^{\tau \log\left(\frac{\pi_0}{\pi_1}\right) + f_0(x) - f_1(x)}\right] \\
&= \alpha_1 \cdot \log\left[1 + e^{-\left(\tau \log\left(\frac{\pi_1}{\pi_0}\right) - f_0(x) + f_1(x)\right)}\right] \\
&= \alpha_1 \cdot - \log \sigma\left(z + \tau \log\left(\frac{\pi_1}{\pi_0}\right)\right) && z = f_1(x) - f_0(x)
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}_{LA}(0, f(x)) &= \alpha_0 \cdot \log\left[1 + e^{\tau \Delta_{01} + f_1(x) - f_0(x)}\right] \\
&= \alpha_0 \cdot \log\left[1 + e^{\tau \log\left(\frac{\pi_1}{\pi_0}\right) + f_1(x) - f_0(x)}\right] \\
&= \alpha_0 \cdot \log(\left[1 + e^{-\left(-\left(\tau \log\left(\frac{\pi_1}{\pi_0}\right) + f_1(x) - f_0(x)\right)\right)}\right] \\
&= \alpha_0 \cdot - \log \sigma\left(-\left(z + \tau \log\left(\frac{\pi_1}{\pi_0}\right)\right)\right) \\
&= \alpha_0 \cdot - \log\left[1 - \sigma\left(z + \tau \log\left(\frac{\pi_1}{\pi_0}\right)\right)\right] && \sigma(-x) = 1 - \sigma(x)
\end{aligned}$$

A.4 LDAM loss

B Tables/Figures

Table 3: Test Results Summary across algorithms and loss functions using PR-AUC evaluation metric

Algorithm	**Log loss**	**LDAM**	**Focal loss**	**LA**
CatBoost	0.319	0.313	0.214	0.178
XGBoost	0.316	0.261	0.327	0.296
LightGBM	0.304	0.254	0.296	0.293

**Minor differences may be due to non-exhaustive hyperparameter search.*

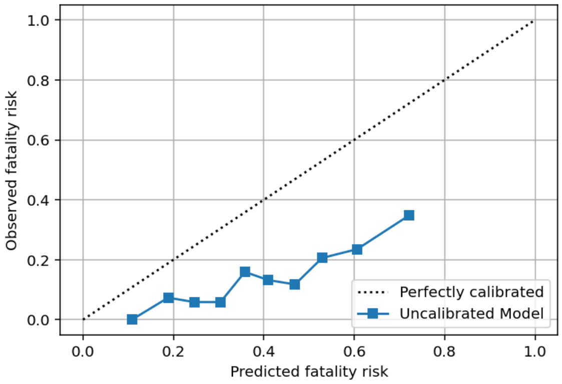


Figure 4: Uncalibrated model on validation set. The uncalibrated model outputs exhibited a general monotonic increasing trend with observed fatality risk, roughly satisfying the logistic assumption for Platt scaling.

B.1 Decision Threshold Optimization

Since PR-AUC is threshold-invariant, the optimal decision threshold was determined post-hoc using Matthews Correlation Coefficient (MCC). Among metrics tailored to imbalanced datasets (e.g. F_1, Balanced Accuracy, Cohen's Kappa, Brier), MCC was favoured for its holistic consideration of classification cases (TP, FP, TN, FN)[¶¶] and stability despite class imbalance.[42]

Table 4: Test Results Summary for Final CatBoost LDAM Model

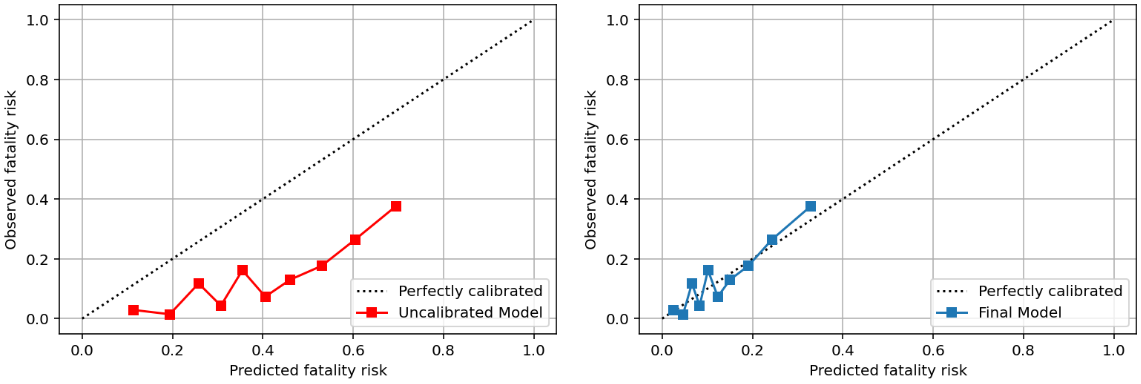
ROC	**PRAUC**	**Acc**	**BAcc**	**Precision**	**Recall**	**F_1**	**MCC**
0.612	0.318	0.811	0.612	0.327	0.337	0.332	0.222

**Threshold = 0.614*

Table 5: Confusion Matrix for Final CatBoost LDAM Model

True	**Predicted**	
	Non-Fatal	**Fatal**
Non-Fatal	527	62
Fatal	60	35

[¶¶] True Positive, False Positive, True Negative, False Negative



(a) Uncalibrated

(b) Calibrated

Figure 5: Comparison pre and post calibration on test set