Predicting the Probability and Cost of Accidents in Austin

Ishrak Wasif Udoy

Muzaffar Yezdan

Shyam Patel

Sanchal Nachappa

Franco Salinas

Agenda

1. Relevance of Problem and Solution Urgency
2. Data Description
3. Baseline Model: Hurdle Model
4. LightGBM
5. Tweedie XGBoost

What's the problem?

Motor-vehicle crashes cost the local government approximately 35.24 million USD in 2022 dollars

One reportable crash occurred every 57 seconds

Identifying when and where crashes are most likely to occur can therefore help design targeted mitigation strategies.

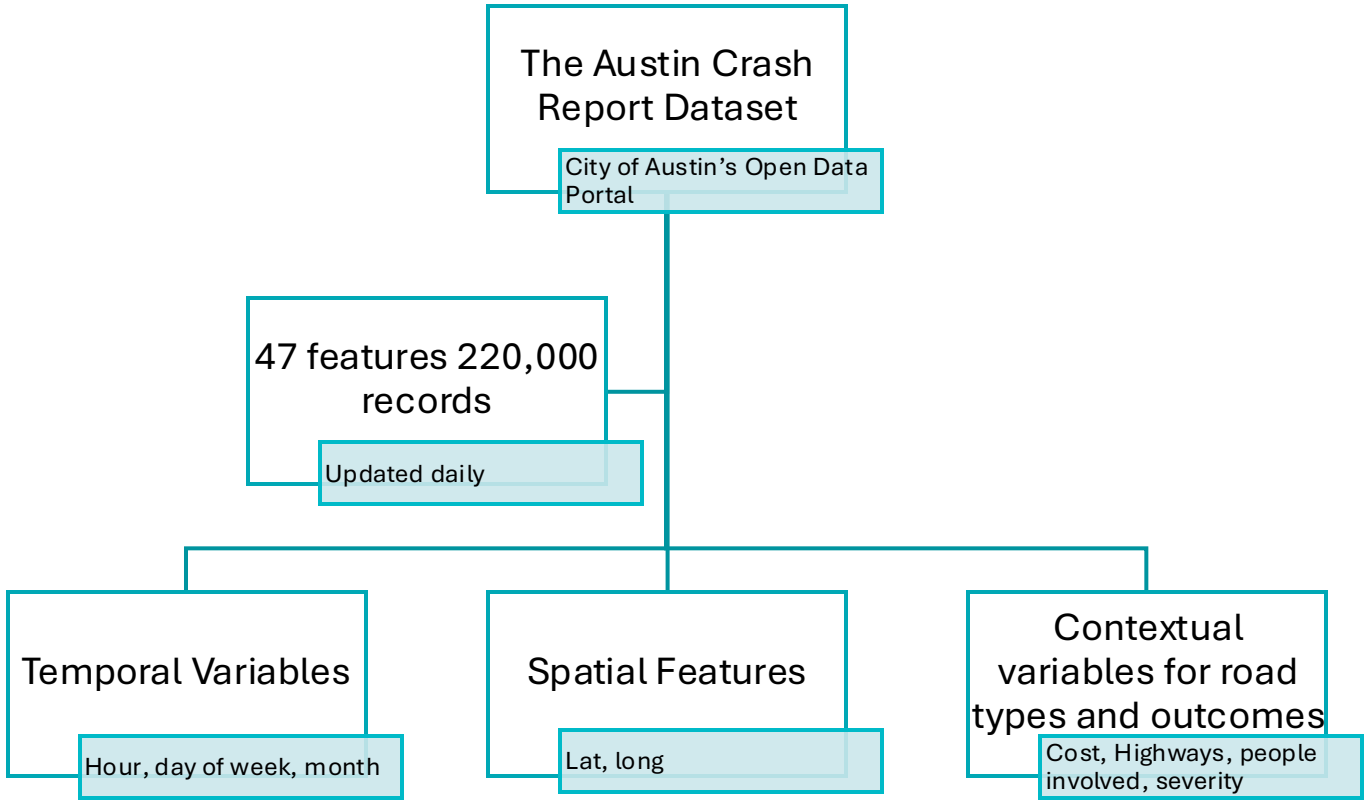
What's our solution?

Build **a spatiotemporal risk surface** that quantifies the **probability and expected cost** of accidents across Austin

Applications:

- Insurance premiums
- Rideshare platforms **compensation** systems

The Data



Engineered Features

- **Time features:** Hour, day, month, holiday/weekend indicators.
- **Spatial features:** H3 grid of resolution index 8- and 1-hour bins
- **Count Features:** Lagged counts, rolling means and recent-incident indicators
- **Weather:** Rain in mm and mean temp in Fahrenheit
- **Negative Observations:** Merge a full hex-time grid for every cell and time bin into the crash data.

Binary LightGBM

Brief Description of the Algorithm

1. Prediction Task:

Binary classification problem: $\hat{p} = P(\text{crash} = 1 | \text{ engineered features})$

2. Light GBM trains an ensemble of decision trees, each one correcting the mistakes of the previous one using the gradient of the loss function.

Binary loss function: $\mathcal{L} = -[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})]$

For each row in the training data $g_i = \frac{d\mathcal{L}}{d\hat{y}_i} \rightarrow$ this tells how wrong was the previous prediction

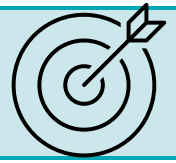
LightGBM fits a new decision tree to approximate this gradient. This process continues until early stopping detects no further improvement on validation data.

3. Ranking Future Hex-Timebins :

We use a Sigmoid function to compute the predicted risk score for every future hex-timebin

Then we sort them from highest $\hat{p}$ to lowest and we flagged the top 5%

Model Specification



Goal:

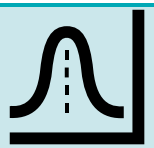
- Predict whether a crash will occur

Predictors:

- 16 meaningful predictors, including rain, temperature, hour of day, month, a rush-hour flag, day-of-week indicators, weekend flag, location group, and speed limit

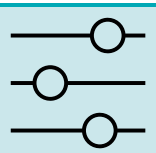
Time-based Splits:

- Train(75%)/Val(15%)/Test Split(15%) to prevent leakage and keep causality



Negative sampling:

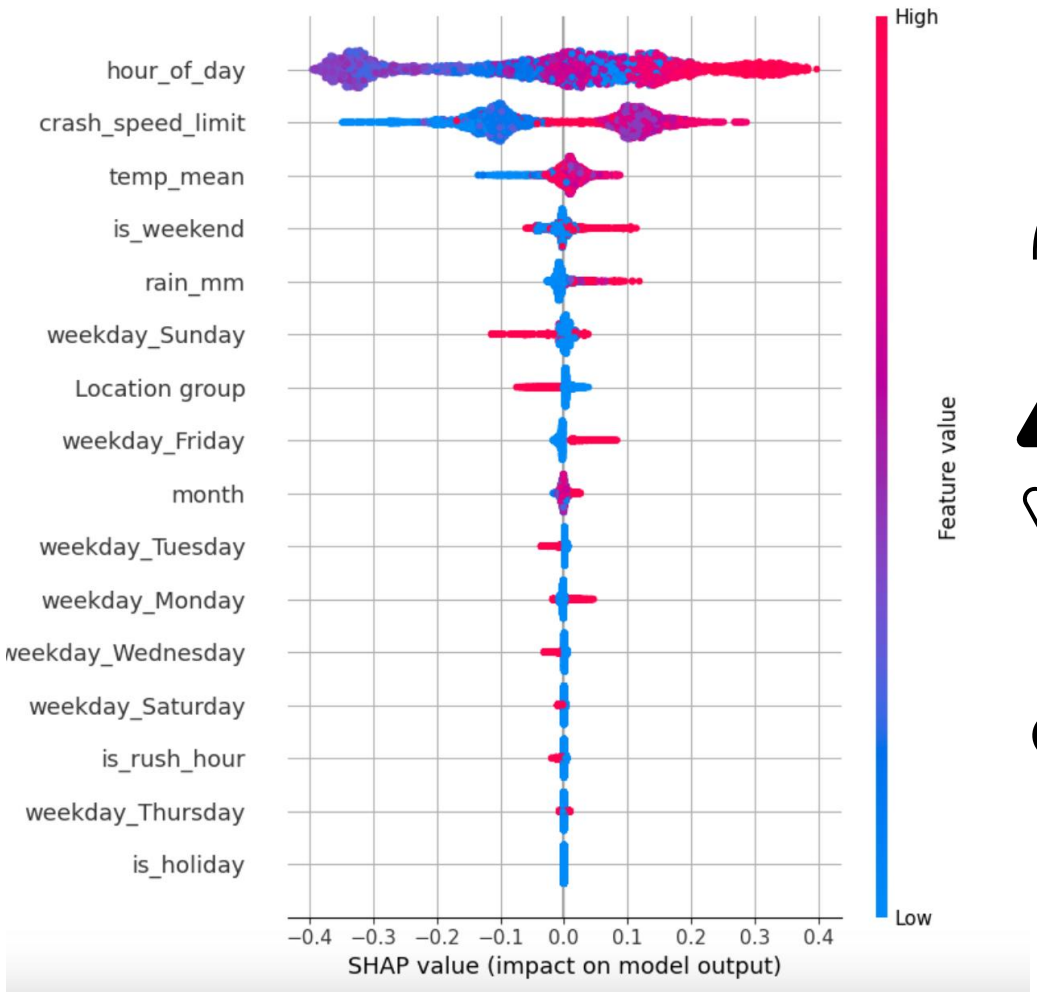
- Down sampling negatives in training (5:1); controlled sampling in evaluation (20:1)



Hyperparameters:

- Small learning rate, 31 leaves, row wise boosting, 0.8 feature and bagging fractions

Results



Crash risk peaks during busy daylight/evening hours and drops overnight

Areas with higher posted speeds contribute meaningfully to crash likelihood

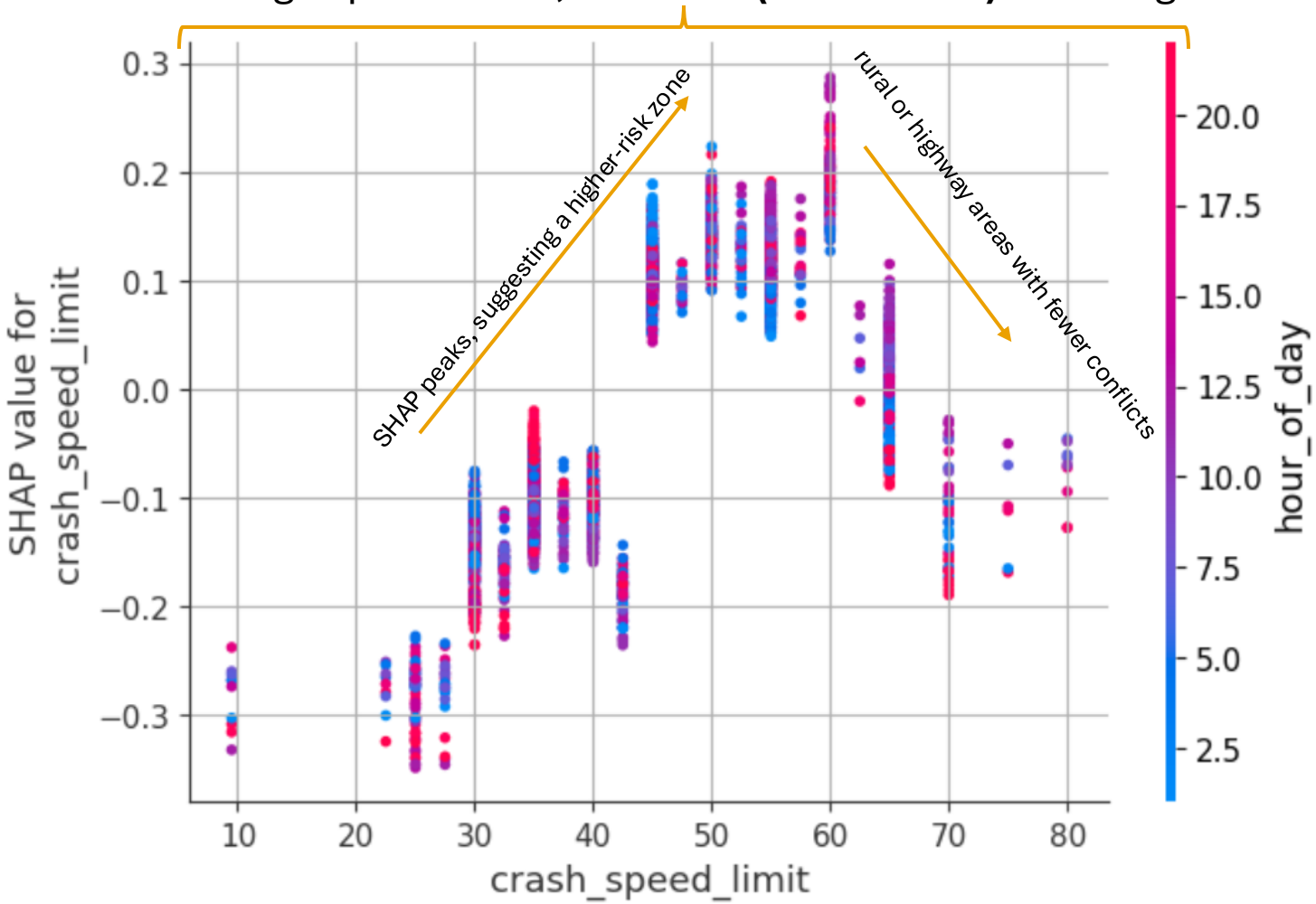
Warmer temperatures slightly increased risk. Crash patterns correlate with seasonality and daytime temperature cycles

Weekend conditions moderately elevate crash probability.

Rainfall is a strong, intuitive driver of crash likelihood

At higher speed limits, the model tends to assign higher crash risk, especially during busy daytime hours

At medium/high speed limits, **red dots (later hours)** show higher crash risk



Performance Metrics

On top 5% bins (>22.81% risk)	
Eval Test set Metrics	**Value**
ROC AUC	64.69%
PR AUC	7.35%
Precision	8.47%
Recall	8.91%

In rare-event modeling, **raw precision/recall numbers look small**, but **the lift over baseline** is what demonstrates value. Our model lifts the crash concentration

"**AUC values in the 0.6–0.7** range are common and **acceptable.**" -Yu & Abdel-Aty (2014)

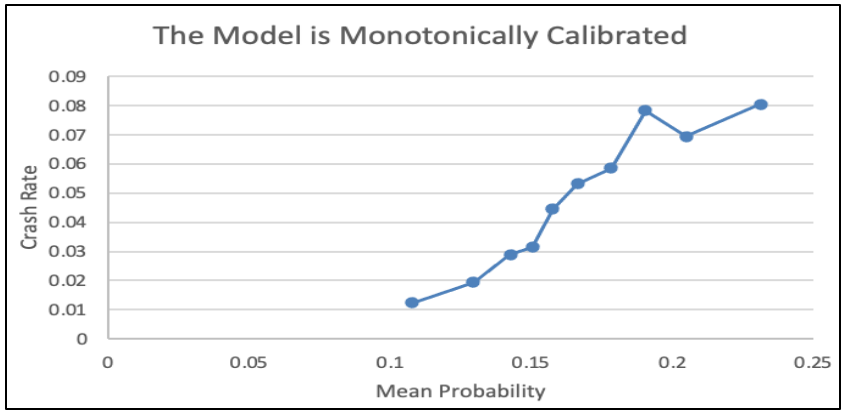
1.53x relative to random 4.8%

1.76x improvement (baseline 4.8% prevalence)

1.78x improvement (baseline 5% by design)

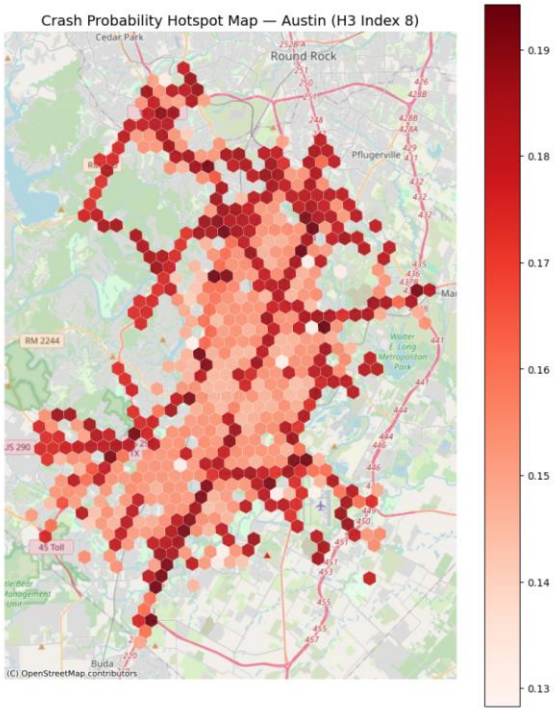
Full Test set Metrics	Value
ROC AUC	64.76%
PR AUC	0.3%

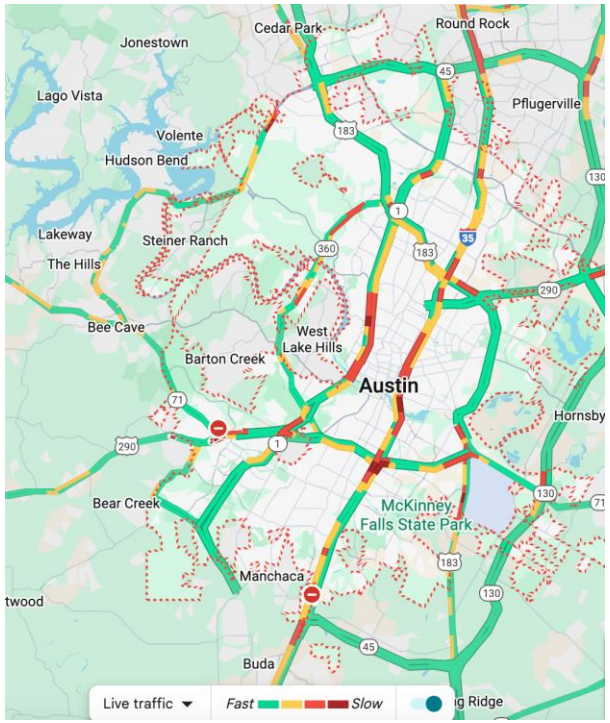
Small PR-AUC because dominated by precision and with 0.2% positive cases the AUC collapses toward the prevalence.



This means the model's ranking is **well-calibrated** as high predicted risk buckets contain more crashes, on average.

This map shows the model's predicted crash risk. Each hexagon represents an H3 cell in Austin, and the color intensity reflects the average crash probability predicted by our LightGBM model over the test period.
The hotspots you see are where the model believes crashes are most likely to occur next, not just where they happened before.





- High-risk regions overlap with historically dangerous corridors like I-35 and central Austin.
- The model also elevates risk in locations where the conditions resemble high-risk patterns even if historical crash counts were low.

Hurdle Model

Brief Description of the Algorithm

Stage 1 — Crash Occurrence (LightGBM Classifier)

- Objective: **binary**
- Key params: LR **0.05**, 31 leaves, feature/bagging frac **0.8**
- Max **2000** trees + early stopping (100 rounds)
- Output: **P(crash > 0)**

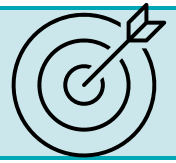
Stage 2 — Crash Count Given Occurrence (LightGBM Poisson Regressor)

- Objective: **poisson**
- Key params: LR **0.05**, 63 leaves, feature/bagging frac **0.8**
- Max **2000** trees + early stopping (100 rounds)
- Trained on **crash > 0** rows
- Output: **E[count | crash > 0]**

Final Prediction

$$\lambda = P(crash > 0) \times E[count \mid crash > 0]$$

Model Specification



Stage 1 Objective:

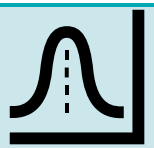
- Binary (for the crash occurrence model)

Stage 2 Objective:

- Poisson (for the conditional crash count model)

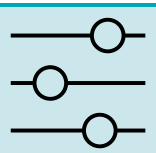
Learning Rate:

- 0.05 (used in both models to control the step size at each iteration)



Number of Estimators with Early Stopping:

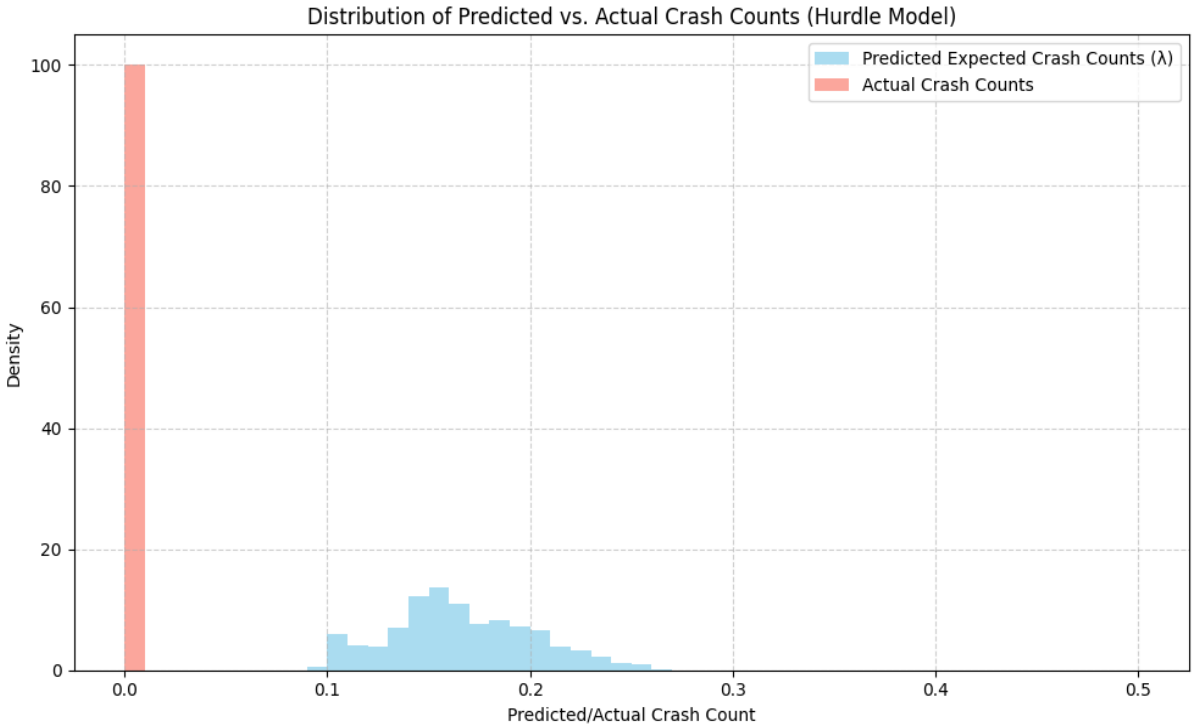
- 2000 with stopping_Rounds=100



Tree Complexity (num_leaves):

- This parameter controls the maximum number of leaves in one tree. It was set to 31 for the binary model and 63 for the count model

Key Insights



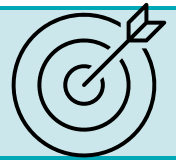
- **Actual crash counts** are almost always zero → confirms extreme rarity
- **Predicted λ values** are small but consistently non-zero due to the hurdle structure
- **Predictions appear inflated** because the binary stage (crash/no-crash) is trained with down sampling
- λ values function as **relative risk indicators**, not calibrated crash count forecasts
- Model is best used for **ranking high-risk locations**, not predicting absolute crash frequencies

Tweedie XGBoost

Brief Description of the Algorithm

- XGBoost with **Tweedie loss**, tuned Tweedie variance power
- Uses **Tweedie loss** which is ideal for **non-negative**, **right-skewed**, highly **heavy-tailed** cost data
- Supports variance power **1 < p < 2**, covering distributions between **Poisson and Gamma**
- Well-suited for **insurance-style severity modeling**, where most crashes are inexpensive, but a few are extremely costly
- XGBoost captures **non-linear interactions** between speed, traffic conditions, weather, and temporal patterns
- **Regularization (L1 + L2)** stabilizes predictions, especially for rare severe crashes
- Naturally handles **missing values**, sparse crash patterns, and high-dimensional feature sets

Model Specification



Goal:

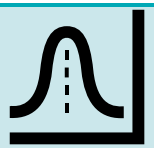
- Predict **Comprehensive Crash Cost** at the hex-hour level, capturing both **economic damage** and **quality-of-life impact**.

Predictors:

- Over **20 engineered predictors**, including rain, temperature, humidity, cloud cover, H3 location, hour-of-day, month, weekend/holiday flags, speed limit, lagged crash cost, and recent-incident (Hawkes-type) features.

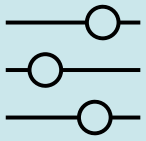
Time-based Splits:

- Temporal split into **Train (75%) / Validation (15%) / Test (15%)** to preserve causality and avoid forward-looking leakage.



Cost Stabilization:

- Applied **log-transform + clipping** to extreme crash costs to reduce heavy-tail volatility and improve Tweedie fit.

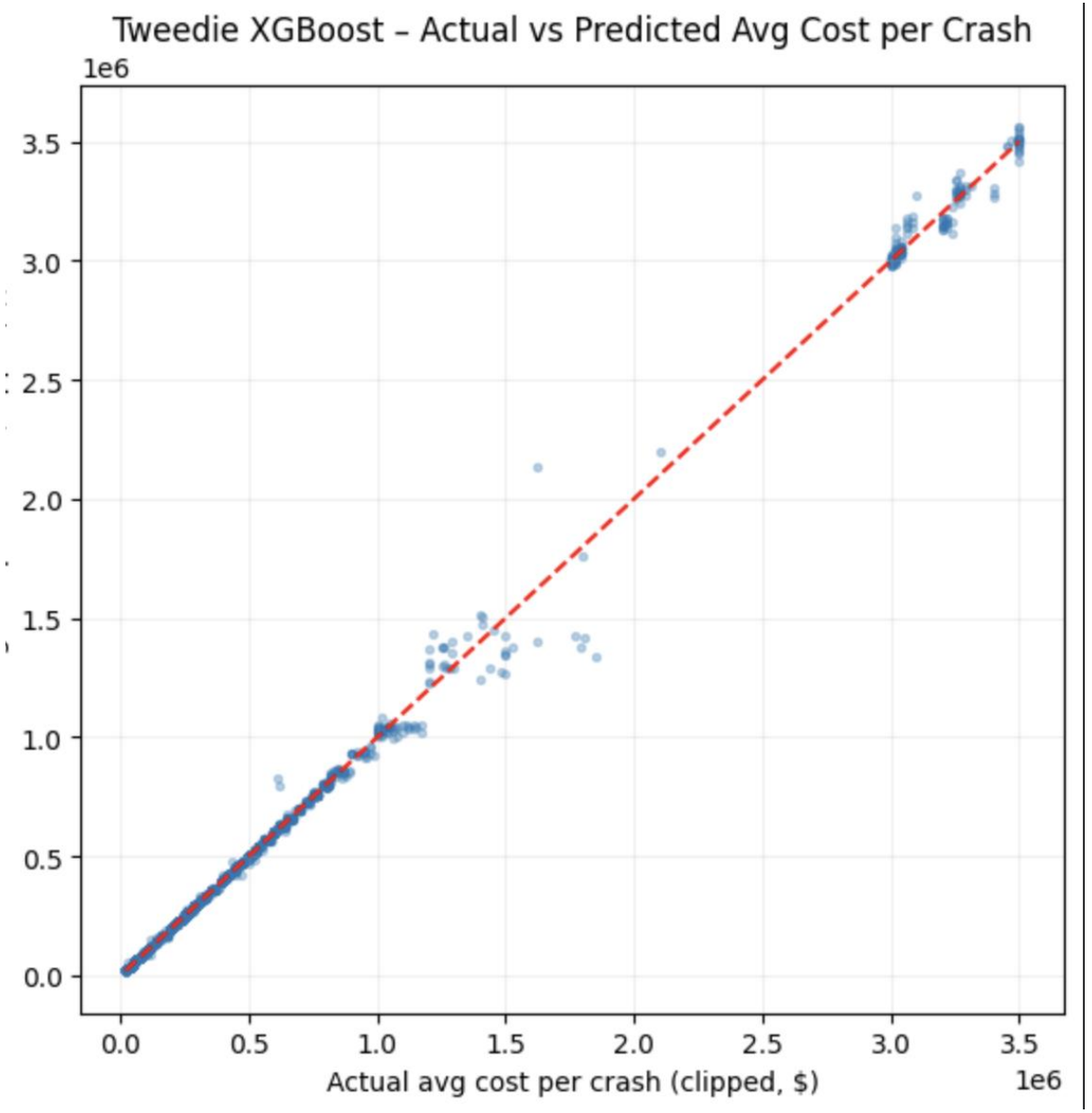


Hyperparameters:

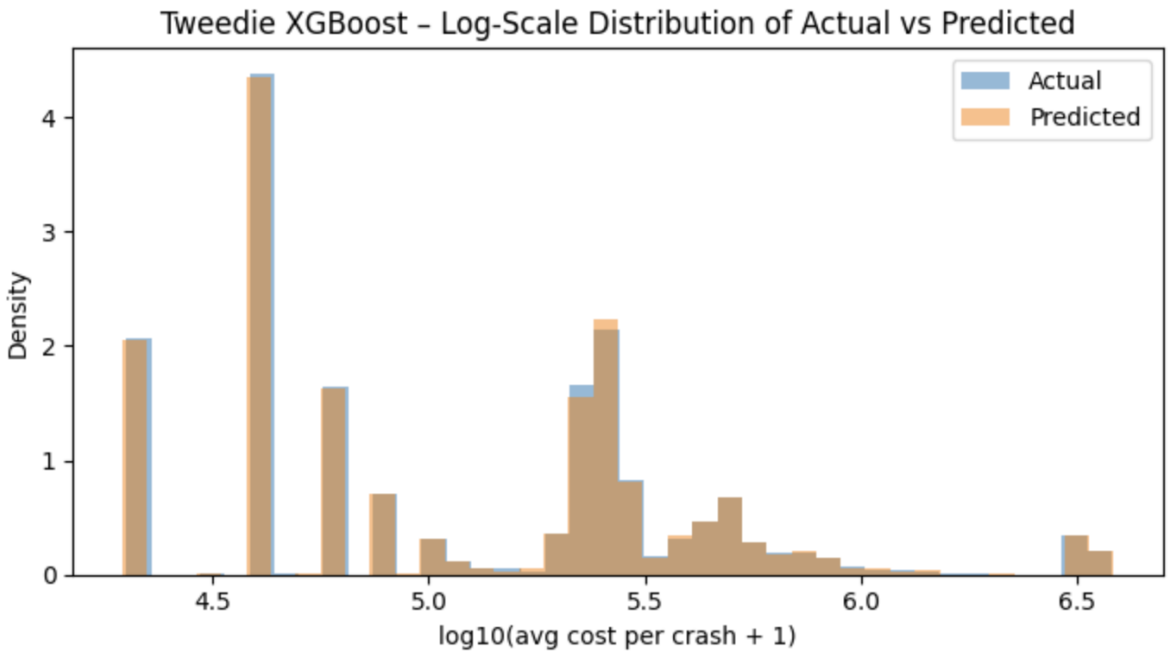
- Tuned **Tweedie variance power p = 1.2**, small learning rate, depth-limited trees, subsampling of rows and features, and L1/L2 regularization for stability on rare high-severity crashes.

Results

- The model predicts **typical crash costs** (e.g., $40k–$120k) with very high accuracy
- Performance on **high-severity crashes** ($250k–$500k) remains strong despite heavy-tailed noise
- Final test metrics: **RMSE ≈ $15.7k**, **MAE ≈ $2.45k**
- Tweedie loss with **p = 1.2** provided the best fit to the skewed cost distribution
- Log-transform + temporal splitting ensured **stable generalization** across different seasons
- Weather adds **contextual improvements**, especially in rain or low-visibility conditions
- Predictions closely track real crash costs across a wide range of severity levels



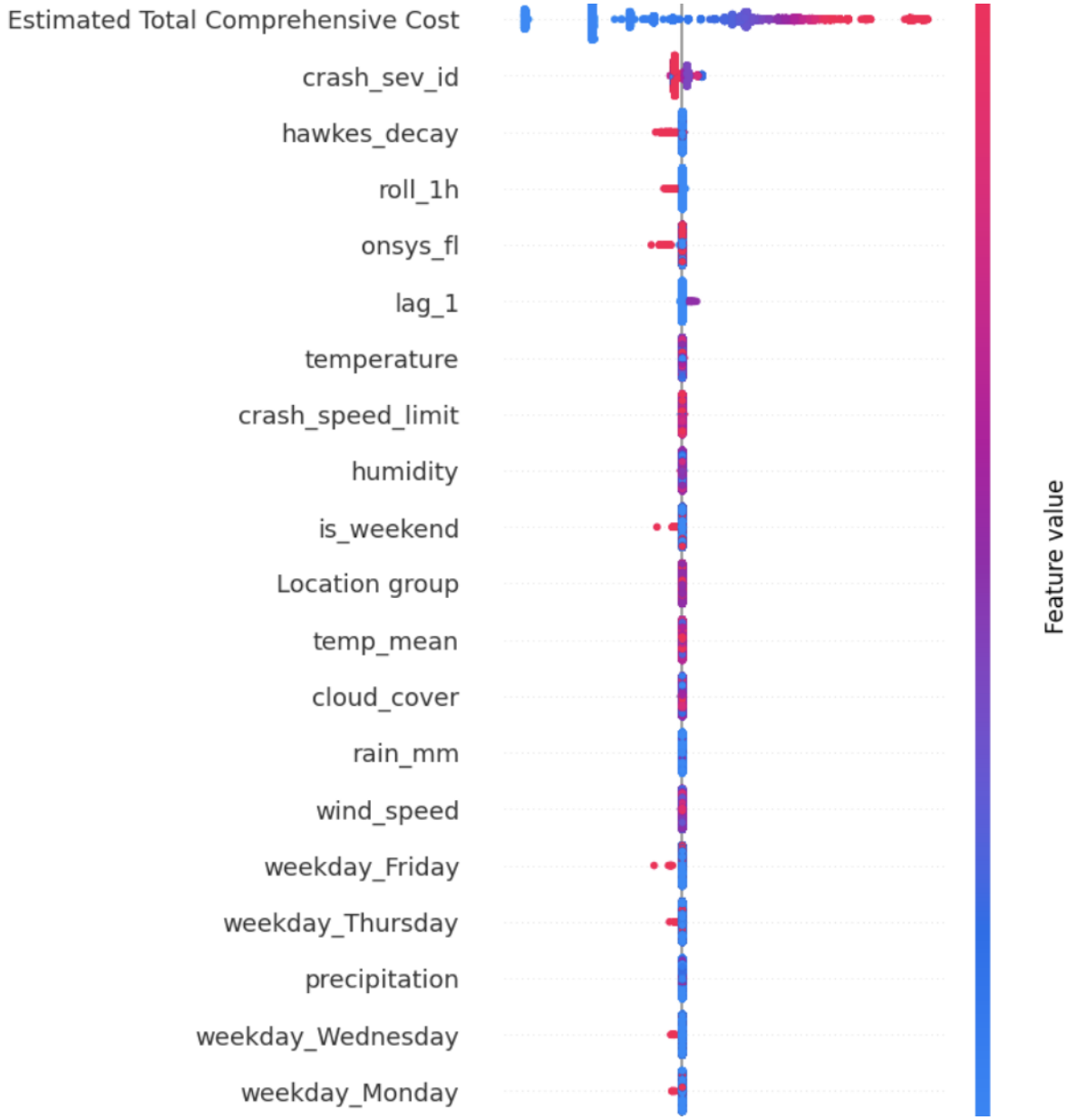
Cost Distribution & Model Fit



- Crash cost data is **extremely right-skewed**, with rare events exceeding $300k–$500k
- The Tweedie model captures this heavy-tailed structure **without collapsing predictions**
- Log-transforming the target stabilizes variance and reduces the impact of extreme outliers
- Predicted values closely follow the **log-scale pattern** of the true distribution
- The model reproduces both **common low-cost crashes** and **high-severity tails**
- Error increases naturally for severe crashes, but the overall distributional shape is well matched
- Confirms that Tweedie XGBoost is appropriate for modeling **insurance-style crash cost behavior**

Model Explainability (SHAP Summary)

- SHAP values identify which features most influence predicted crash cost
- **Lagged severity** and **recent-incident indicators** contribute most to high-cost predictions
- **Speed limit** and roadway context strongly impact expected severity in faster corridors
- **Weather conditions** (rain, temperature, humidity) add meaningful adjustments during adverse periods
- **Temporal patterns** (hour, month, season) capture cyclical severity trends across the year
- Confirms the model is learning **realistic, interpretable severity drivers**, not noise



Key Takeaways & Next Steps