Catcher Framing Evaluation

Joe DiGiovanni

Executive Summary & Methodology

This project uses machine learning to measure a catcher's ability to frame strikes, isolating the catcher from umpire and pitcher habits. We used a probability-based boosting model to calculate strikes cumulatively added by each catcher over multiple seasons. We focus on the shadow zone to ensure catchers are scored for framing balls as strikes, while ensuring they do not receive true strikes as called balls.

To isolate catcher ability, catcher, umpire, and team IDs were excluded. This prevents the model from learning biases on categorical variables and prevents data leakage. This creates a league average baseline, separating framing ability from historical ability. If the model determines IDs correlate with framing, it will struggle to generalize to unseen data.

Feature Engineering

Ball height was normalized relative to the zone's upper and lower boundaries to account for different batter heights. Pitch-level movement (velocity, horizontal/vertical break) is included; different pitch types are received differently and land in different parts of the zone. Framing a high fastball requires different skills than a slider. Since we lack 3D catch coordinates, we cannot measure the exact distance between the plate and reception. Future data could incorporate this, replacing pitch characteristics as a proxy.

Model Selection

The HistGradientBoostingClassifier was chosen for efficiency, accuracy, and interpretability. The strike zone is theoretically a 3D volume, but practically a blob. Thus the variable nature of the zone allows the model to learn non-linear boundaries and interaction effects (e.g., "low-and-away" is called differently for Lefties vs. Righties). The model runs on scikit-learn 1.3.2, limiting dependency issues and promoting reproducibility.

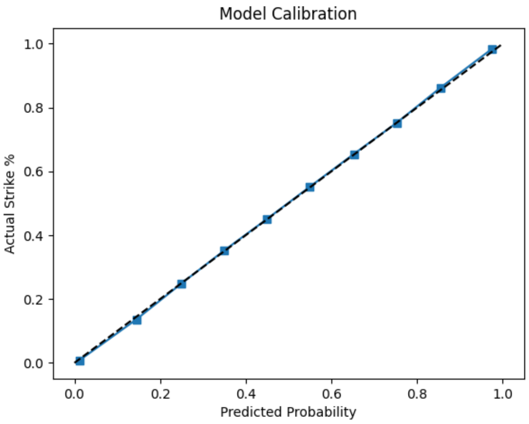


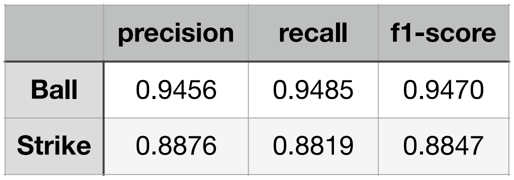
Figure 1: *Reliability Diagram (Calibration Curve)*

Validation & Data Hygiene

The model was validated on a strict holdout dataset using Group Shuffle Split on Game IDs. By evaluating on unseen games rather than random pitches, we mitigate data leakage from game-specific conditions or umpire tendencies. Rigorous data cleaning was applied to the training set to filter sensor errors (extreme outliers in height or velocity), ensuring the model learns from physical reality rather than tracking artifacts.

Validation & Performance (Table 1)
The final model achieves strong classification metrics on the holdout data:

- **Accuracy:** 92.74% global accuracy.
- **Log Loss:** 0.1655 (indicating high confidence in probabilistic outputs).
- **Precision/Recall:** The model achieved a Precision of ~0.89 and Recall of ~0.88 for Strike predictions, confirming that the "Strikes Added" metric is robust and minimizes false positives



	precision	recall	f1-score
Ball	0.9456	0.9485	0.9470
Strike	0.8876	0.8819	0.8847

***Table1:** Performance Metrics*

Model Reliability
To ensure the metric is fair, we analyzed the model's calibration (Figure 1). The reliability diagram confirms the model is well-calibrated; a pitch assigned a 70% strike probability is indeed called a strike ~70% of the time historically. This ensures our "Strikes Added" values serve as a fair accounting of runs saved without inflating values on impossible-to-frame pitches.

Spatial verification (Figure 2) confirms the model has "learned" the realistic strike zone—correctly identifying the rounded corners and the expanded horizontal edges typically granted by umpires—without rigid hard-coding.

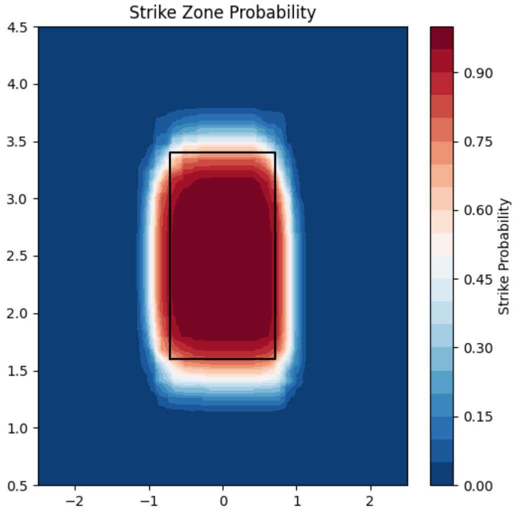


***Figure 2:** Learned Strike Zone Heatmap (Probability Surface)*