Caleb Lee

Runner Advance Prediction (Problem 1)

Objective: Predict runner_advance as a calibrated probability in 0,1 for sacrifice-play opportunities and submit probabilities for runner_advance_predictions.csv. The Primary evaluation is log loss; the secondary is the soundness of method.

My process, in order

1. **Frame the target and metric**
   Target is binary: 1 if the runner advanced, 0 if held or thrown out. The primary metric is log loss because we care about well-calibrated probabilities, not just classification accuracy.

2. **Audit the data and plan for leakage**

   - **Train**: 15,533 labeled opportunities.

   - **Tracking**: 62,574 outfielder tracking rows at sub-play resolution. Multiple tracking samples exist per play.
     To avoid overweighting plays with more tracking rows and to prevent leakage from post-contact tracking detail, I aggregated tracking to one row per play_id (numeric mean, min, max) and then broadcast those per-play features to each runner opportunity on that play.
     I removed any outcome-like columns other than the target, following the glossary.

3. **Build features that match baseball logic**

   - **Context**: inning, score differential, balls, strikes, outs.

   - **Situation**: runner_base, runner_on_3rd, two_outs, high_leverage, plus focused interactions such as runner_on_3rd × two_outs.

   - **Ball off the bat**: exit speed, launch angle, direction, hang time, distance, and simple buckets (ground ball, line drive, fly ball, popup; weak to very hard; shallow to deep).

   - **Positioning from tracking**: outfield depth and distance to home, plus aggregated pos_x and pos_y summaries (mean, min, max).
     Total used in modeling: 47 features, including 8 from aggregated tracking.

4. **Decide how to validate**
   Plays from the same game are not independent. I used 5-fold GroupKFold on game_id so entire games stay inside a single fold, which prevents within-game

leakage across trains and validation.

5. **Establish baselines before modeling**
   - **Constant prevalence**: 0.6931 log loss (p = 0.499).
   - **Out-of-fold contextual baseline** using only runner_base, outs, and hit_trajectory with the same grouped CV: 0.5907 ± 0.0050.
     This gives a fair "smart guess" reference that my model needs to beat.
6. **Model search and choice**
   I compared models with the exact same grouped CV and neg_log_loss scoring.
   - **HistGradientBoosting**: 0.3269 ± 0.0220 (best)
   - **RandomForest**: 0.3683 ± 0.0126
   - **LogisticRegression**: not competitive for the non-linear structure
     I chose HistGradientBoosting because it is strong on structured tabular data with mixed types, captures non-linear interactions, and has built-in regularization.
7. **Probability quality and calibration**
   I tested isotonic and sigmoid post-hoc calibration with the same grouped CV. Neither improved out-of-fold log loss, so I kept the uncalibrated HGB. The predicted probabilities have healthy spread (mean 0.521, SD 0.363), and reliability looked acceptable for decision use.

Result and how to read it:

- **Final cross-validated log loss**: 0.3269.
- Improvement vs constant baseline: absolute 0.3662, which is about a **53%** relative reduction (0.6931−0.3269)/0.6931(0.6931 − 0.3269) / 0.6931(0.6931−0.3269)/0.6931.
- Improvement vs contextual baseline: absolute 0.2638, which is about a **45%** relative reduction (0.5907−0.3269)/0.5907(0.5907 − 0.3269) / 0.5907(0.5907−0.3269)/0.5907.
- Signals that matter line up with baseball intuition: **runner_base**, ball-flight physics (**distance, angle, exit speed**), and **outfield depth/positioning** from tracking.

Reproducibility and Deliverables

- Code: notebook and script implement the full pipeline, including tracking aggregation, feature build, GroupKFold CV, model training, and export.

- Predictions: runner_advance_predictions.csv (probabilities for test_data.csv).

Next Steps with More Time and Data

1. **Richer context:** park effects, weather, batter spray tendencies, and fielder arm/relay timing.

2. **Entity modeling:** partial pooling or embeddings for runners, hitters, and fielders to stabilize sparse groups.

3. **Calibration at scale:** fold-wise stacked calibration or constrained reliability optimization; report reliability curves and Brier score.

4. **Temporal robustness:** rolling backtests by month/season to monitor drift and stability.

5. **Explainability:** global and local effect estimates for coaches and analysts (for example SHAP grouped by feature families).