Caleb Lee

Runner Advance Prediction (Problem 1)

Objective:

Predict runner_advance ∈ [0,1] on sacrifice plays and supply calibrated probabilities for test_data.csv, in terms of log loss, with brief, reproducible workflow.

Data & leakage management:

- Train (15,533 rows): sacrifice plays with runner_advance (1 = advanced; 0 = no advance or failed).
- Tracking (62,574 rows): outfielder positions per play. In order to avoid plays with numerous tracking samples from overwhelming the computation, I aggregated tracking to one row per play_id (mean/min/max per numeric column) and joined onto train/test. Since there are usually several runners who have opportunities on one play, I did not drop training rows that have the same play_id but broadcast the aggregated features to all rows.
- Glossary guided leakage checks; I dropped any outcome-like columns other than the target.

Feature engineering:

Context (inning, score diff, balls/strikes), situation (runner_base, runner_on_3rd, two_outs, high_leverage, interactions), physics of batted-balls (exit speed, launch angle/direction, distance; discretized bins), and tracking-derived proxies (outfield_depth, dist_to_home, plus agg mean/min/max from tracking).

Validation & baselines:

CV protocol: 5-fold GroupKFold on game_id in order to prevent within-game leakage.

Baselines:

- Constant-prevalence (p = 0.499): 0.6931 log loss.
- Out-of-fold contextual baseline (grouped on runner_base, outs, hit_trajectory): 0.5907 (±0.0050).

Models:

I trained three families: Random Forest, Histogram Gradient Boosting (HGB), and Logistic Regression. HGB performed most accurately in detecting nonlinear interactions in tabular data with intrinsic regularization.

- RandomForest: 0.4046 (±0.0120)
- HistGradientBoosting: 0.3882 (±0.0132) ← best
- LogisticRegression: 0.4762 (±0.0137)

Calibration:

I contrasted isotonic and sigmoid calibration on identical grouped CV. Both decreased slightly OOF log loss (isotonic 0.4041, sigmoid 0.4045), so I retained the uncalibrated HGB for final prediction. (Reliability/Brier on request; probabilities had good spread: mean 0.521, SD 0.363.)

Result & interpretation:

The final model has 0.3882 log loss, a ~44% improvement on constant (0.6931) and ~35% on OOF contextual (0.5907). The top predictors are in line with baseball intuition: runner_base, hit distance/angle/exit speed, and outfield depth/positioning.

If given more time + resources:

- Add context (park, weather, batter spray, arm/relay timing).
- Hierarchical shrinkage priors per-entity (runner, hitter, fielder).
- Calibrate using fold-wise ensembling or post-hoc reliability optimization; include reliability curves + Brier in the appendix.
- Temporal backtests (month/season) to ensure stability.

Deliverables:

Code: mariners_analysis.py (reproducible), with grouped CV and OOF contextual baseline.

Predictions: runner_advance_predictions.csv (probabilities for test).