The Intent to Give it Your All

SMT Data Challenge 2025

https://github.com/kimdoyo5/SMT2025

Abstract

Modern player evaluation concentrates on talent metrics such as maximum pitch velocity and sprint speed due to their predictive capacities. However, the expression of talent depends on deliberate athlete **intent**. Despite widespread acknowledgment, the concept of "hustle" remains relegated to conventional wisdom rather than quantitative analysis. In this project, we develop quantitative metrics, **Hustle+** and **Safe Probability Added** (SPA), to objectively measure hustle and its influence in game outcomes. We define these metrics through statistical estimation of athletes' true capacities and their effort-driven expression. True capacity is estimated using softened maximum-a-posteriori estimation combined with Z-score-based variance reduction, while the expression of this capacity is modeled based on established sport science literature. Gradient-boosting decision tree ensembles are employed to simulate outcomes. Results indicate a linear relationship between hustle and performance. Furthermore, significant discrepancies in hustle are identified among teams, with consequential impacts on seasonal outcomes.

TABLE OF CONTENTS

INTRODUCTION 1

1.1 Introduction ... 1

METHODS 2

2.1 Data Preparation ... 2

2.2 Predicting Outcomes ... 2

2.3 Estimating the Runner's "Effort" ... 5

2.4 Hustle+ ... 8

2.5 Safe Probability Added (SPA) ... 8

ANALYSIS 9

3.1 Can We Quantitatively Measure Hustle? ... 9

3.2 Is There Such a Thing as a "Hustle Culture"? ... 10

3.3 Can Teams Actually Win Games by Hustling? ... 10

CONCLUSION 11

4.1 Conclusion ... 11

4.2 Future Work ... 11

ACKNOWLEDGEMENT 12

APPENDIX 13

A. Inferring Throwing Target ... 13

B. Cubic Smoothing Splines ... 14

C. Classifying Catch Outcomes ... 15

D. GBDT Fine-tuning + Ensembling ... 16

E. Bootstrap Aggregation ... 17

F. Softened MAP Estimation ... 18

G. Variance Reduction using Z-scores ... 21

H. Sport Science Literature ... 21

I. Quantile Regression ... 22

REFERENCES 23

INTRODUCTION

1.1 Introduction

Figure 1: "Charlie Hustle" Pete Rose diving towards home.

Every year, coaches across ballparks emphasize the importance of hustle—hustle around the bases, hustle after the ball, even hustle in and out of the dugout.

Despite widespread acknowledgment, the **concept of *"hustle"* remains relegated to conventional wisdom rather than quantitative analysis.**

In contemporary baseball analytics, player evaluations are centered around "talent" metrics, such as *maximum* swing power, throw velocity, and sprint speed; the word "hustle" appears only on rare occasions.

Nonetheless, beneath every powerful swing, hard throw, and a fast sprint there is a **deliberate intent** to exert maximal effort.

Can we quantitatively measure hustle?

Is there such thing as a "hustle culture"?

Can teams actually win games by hustling?

This project aims to answer these questions by quantifying hustle and its influence on game outcomes.

The Scenario of Interest

Consider:

Batter running out an infield ground ball.

This is a classic example of **hustle**. Nearly 90% of fielded ground balls result in outs at 1st base.[1] Hustling in this scenario means taking the *off-chance* that an out becomes an error or a double play becomes a fielder's choice, even though it does not contribute to his on-base percentage.

Hence, we concentrate on *contests at 1st base*, defined as single play sequences featuring (in order):

1. A batted ball bouncing in the infield
2. An infielder acquiring the ball
3. An infielder[2] *intending* to throw a ball to 1st base
4. The batter running to 1st base

In particular, we use two criteria to infer whether the infielder *intends* to throw to 1st base. (See Appendix A.)

[1] In our dataset, 10.79% of fielded ground balls resulted in safe.
[2] This infielder may be a different player—consider double plays, for instance.

METHODS

2.1 Data Preparation

The data used for this project comes from 274 anonymized Minor League Baseball (MiLB) games. It includes 2D player tracking data and 3D ball tracking data at 20/30 fps.[3]

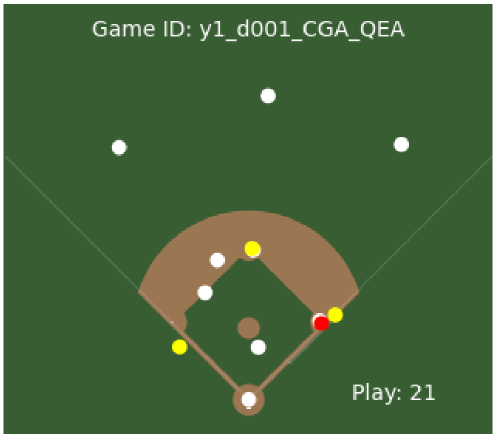

Figure 2: Example play animation.

The datapoints are associated with `game_id`'s, `play_id`'s, and `player_id`'s. However, some data points have missing `id'`s.[4]

After initial filtering (for the scenario of interest), we retrieve 3,483 datapoints. Using an artificial **id** labeling process[5], we retrieve 90 more.

Denoising Position Data

We convert 2-dimensional player coordinates into 1-dimensional by taking the *distance from 1st base*. (Figure 3) This both reduces dimensionality and makes the distance calculation more reliable.[6]

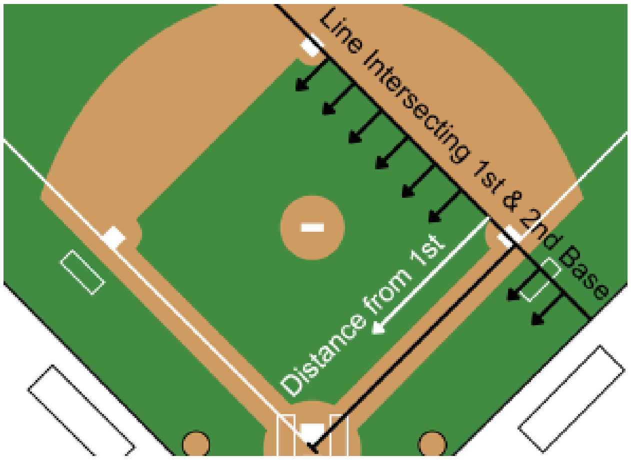

Figure 3: Distance from 1st base, defined as the length of the projection of the coordinates onto the line intersecting 1st and 2nd base.[7]

We calculate the batters' *speed towards 1st base* using the distance over minimum time intervals (50/33ms)[8] to maximize precision. Since the data are extremely noisy[9], we apply **cubic smoothing splines.** (Figure 4) (See Appendix B.)

2.2 Predicting Outcomes

Recall our scenario of interest: *Batter running out an infield ground ball.*

Thus, our outcome of interest is: *Whether the batter was safe at 1st base.*

Since the dataset does not provide play-outcome information, we classify ground balls as `safe` if-and-only-if the batter reaches base[10] in the subsequent `play_id`.[11]

[3] Camera snapshots at 50/33ms intervals.
[4] There are 3837 candidate data points; 354 has missing id's.
[5] See analysis code for details.
[6] When the batter touches first base, he is not precisely running towards it; he runs past it at the side.
[7] Equivalently, this line intersects both 1st and 2nd base.
[8] This is the time interval for 20/30 fps camera snapshots.
[9] Taking a naive displacement over 500ms can make speed estimates off by up to **13-20%** on a 20/30 fps camera.
[10] In the case(s) where the batter attempted to reach other bases, we check every base that he attempted to reach.
[11] The edge case(s) where the batter was thrown out after successfully reached 1st base were manually labeled.

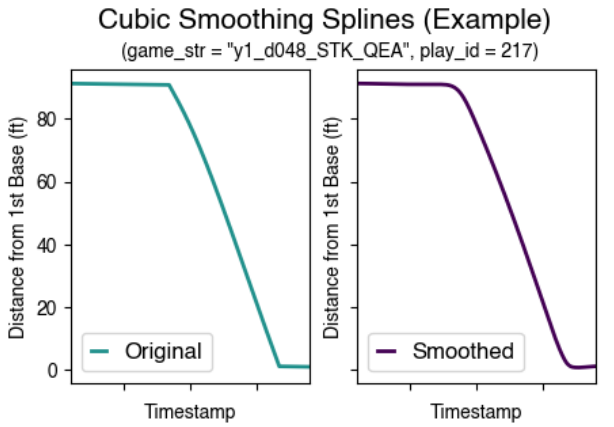

Figure 4: Smoothing pushes the sharp edges (extreme accelerations[12]) towards reasonable ranges.

A **naive approach** to predicting safe(s) uses the batter's position when the 1st baseman catches the ball. In other words,

1. If batter reaches 1st base *before* catch: `Safe`
2. If batter reaches 1st base *after* catch: `Out`

However, this approach assumes that the batter is safe **if-and-only-if** he reaches 1st base *first*, which is **false** if:

1. The 1st baseman does not catch the ball;
2. The 1st baseman drops the ball;
3. The 1st baseman's foot comes off the base;[13]
4. The ball hits the batter.

In other words, the batter may influence outcomes by **influencing the opposition players**. So, we separate the prediction process into 2 steps:

1. **Predicting catch outcomes:** `CatchOn/CatchOff/Dropped/Passed`
2. **Predicting play outcomes** (using catch outcomes and timing): `Safe/Out`

By considering how the batter influences catch outcomes, we separate out the **external factors** (fielders) from **internal factors** (batter).

Also, this improves predictive capacities not only by explicitly accounting for different situations but also preventing **overfitting**.

Predicting Catch Outcomes

"Given the throw & runner, determine the probability of the 1st baseman..."

1. Catching the ball while touching 1st base — `CatchOn`
2. Catching the ball without touching 1st base — `CatchOff`
3. Blocking the ball near 1st base — `Dropped`
4. Failing to keep the ball near 1st base — `Passed`

(To see how the tracking data is used to classify the four catch outcomes.) (See Appendix C.)

The two factors—throw and runner trajectory—are *spatiotemporal* and *high-dimensional*.[14] This makes any statistical model prone to **overfitting**, especially given a small dataset.

We propose a dimensionality-reduction solution: **2D Target Plane** (in 3D space). (Figure 5)

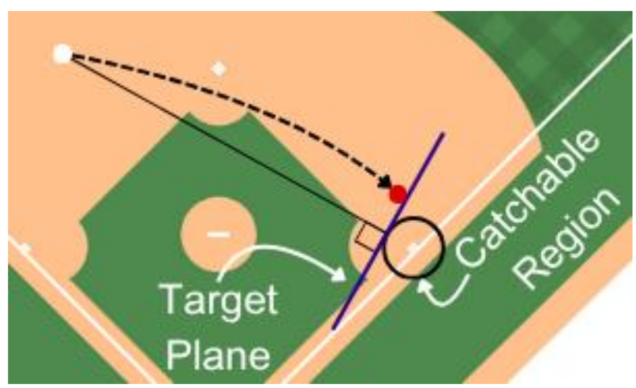

Figure 5: Bird's eye view of the target plane, defined as the plane perpendicular to the line from ball release to 1st base, tangent to a 10ft-radius "catchable region" sphere.

[12] Often upwards of 100mph sprint speeds.
[13] And does not touch the base before the batter does.
[14] A single throw across the diamond has **20+** timestamps of 3-D coordinates (and **20+** timestamps of 1-D batter distance from 1st base).

Thus, each throw is summarized into the following features:

1. The orientation of the target plane
2. The ball's location on the target plane
3. The ball's travel direction (when ball intersects target plane)
4. The batter's distance from 1st base (when ball intersects target plane)

Using these features, we **predict the 4 catch outcomes** using an **ensemble** of gradient-boosted decision tree (GBDT) models: `XGBoost`, `CatBoost`, and `LightGBM` (Figure 6) (See Appendix D.)

Predicting Play Outcomes

We can now predict `Safe/Out` using the outputs:

1. `CatchOn` → `Safe/Out`
2. `CatchOff` → `Safe/Out`
3. `Dropped` → Always `Safe`
4. `Passed` → Always `Safe`[15]

To further remove the external factor (1st baseman) from the outcome[16], we predict the **`expected catch_timestamp`** using (GBDT) **`CatBoost`** fitted on the same features. We also perform **bootstrap aggregation** to reduce variance. (See Appendix E.)

Using **`expected catch_timestamp`**, if the catch outcome is `CatchOn/CatchOff`, we predict Pr(`Safe`) using the **logistic regression model** defined in Appendix C.

Putting it Together

Thus, we have:

$$
\begin{aligned}
\Pr(\texttt{Safe}) = {} & \Pr(\texttt{CatchOn}) \times \Pr(\texttt{Safe if CatchOn}) \\
& + \Pr(\texttt{CatchOff}) \times \Pr(\texttt{Safe if CatchOff}) \\
& + \Pr(\texttt{Dropped}) \\
& + \Pr(\texttt{Passed})
\end{aligned}
$$

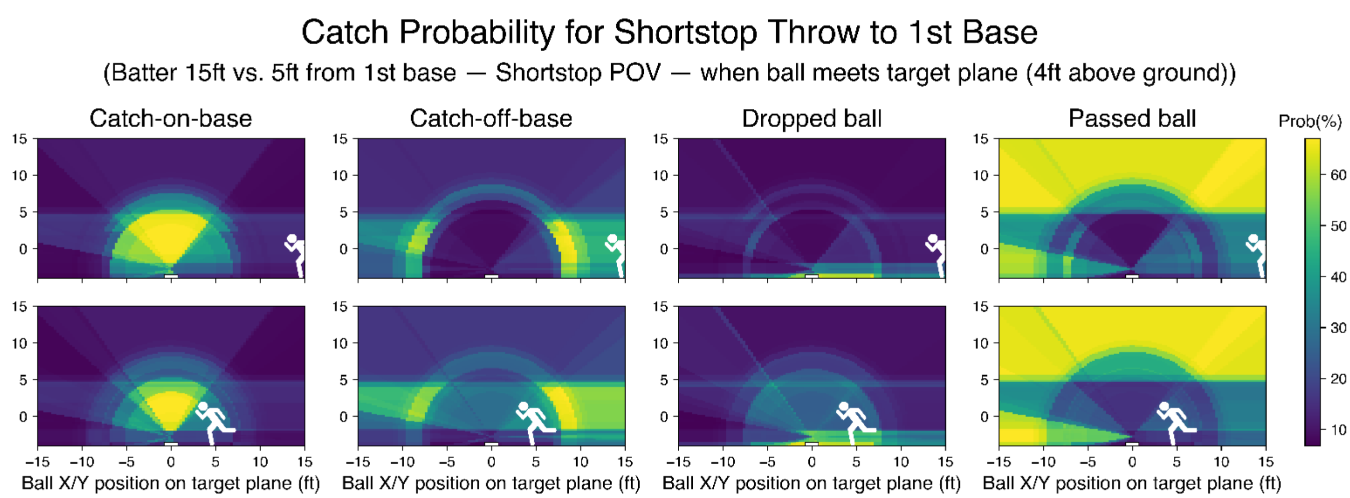

Figure 6: Catch-on-base probability is greatest at the center of the target plane (0,0), which is on average 4ft above ground. As the runner approaches closer to 1st base (at intersection time), catch-on-base probability decreases, while other outcomes increase in probability.[17] This demonstrates the batter's influence on the opposition players.

[15] Possible advances, different from Dropped.

[16] Thereby isolating the batter in our analysis.

[17] The probabilities likely increase due to 1st baseman trying to stretch too far and catch tough throws while fixed on the base.

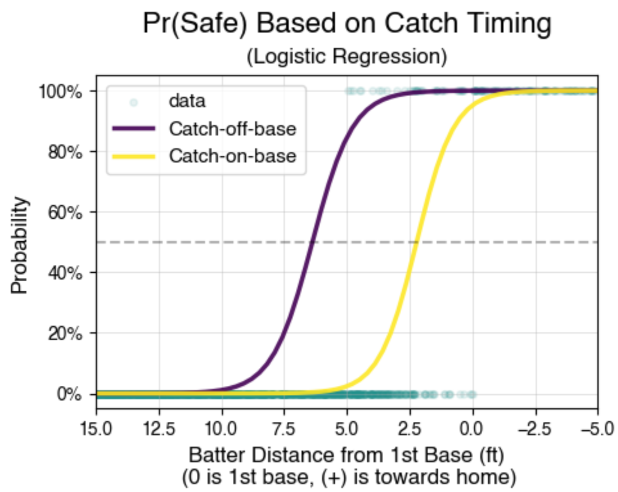

Figure 7: The safe probability on a catch-on-base crosses 50% at around 2.5ft, which suggests that the average batter has a 2.5ft "reach" with his legs. On the other hand, catching off-base significantly increases safe probability.

2.3 Estimating the Runner's "Effort"

We use the following framework for understanding the relationship between *effort* ("hustle") and *performance*:

Performance[18] = True Capacity ⊗ Effort Level

Where ⊗ is an *unknown function* that translates True Capacity to Performance using Effort Level as a parameter.

To estimate the player's effort level, we:

1. Infer the player's *true capacities*;
2. Estimate function ⊗ using existing **sport science literature;**
3. Reverse-engineer *effort level* using the above equation.[19]

Inferring the Runner's "True Capacity"

We posit every player has a **runner signature**: a set of (largely uncorrelated) *true capacities* that determine his running behavior, independent of *effort*, and cannot be changed in the short term:

1. Maximum speed[20]
2. Maximum acceleration
3. Time to start running after ball contact

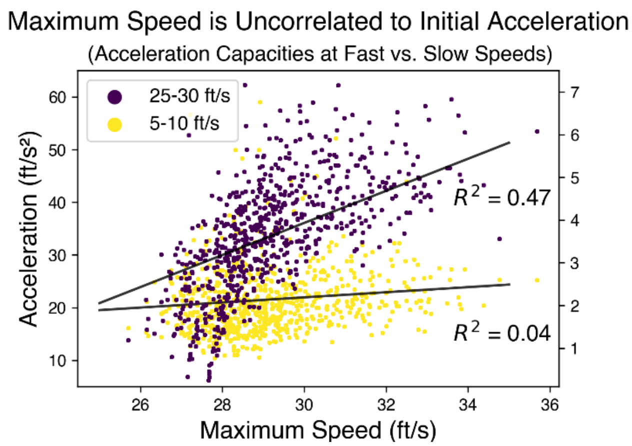

Figure 8: The batter's initial acceleration capacity has little to no correlation with maximum speed. This suggests that acceleration and maximum speed are distinct qualities.

For estimating *maximum speed*, a naive extrapolation of limited data[21] could not work—as **most players did not have enough chances to reach (near) true maximum speed** within the dataset.[22]

Thus, we must *infer* players' maximum speed. We do this by setting players with large (≥5) `run_count` as a *prior* and making **(softened) MAP estimations**. (See Appendix F.)

While each player has more instances of *near-maximum acceleration*,[23] acceleration is a *derivative* of speed, and therefore subject to

[18] In our scenario, Performance is the ability to create *positive acceleration*. Equivalently, the ability to create *force*. (F = ma)
[19] Performance (acceleration) is calculated in Data Preparation.
[20] This can be interpreted as the ability to continue accelerating at a high velocity.
[21] The vast majority of players had less than five run instances.
[22] The batter may have never ran with, throughout the entirety of the run, maximum intent, or optimal running mechanics.
[23] There are many instances where a player reached top acceleration (in parts of the run) but never reached top speed.

substantial noise (despite the Denoising Process). Hence, we alternatively[24] use *95th percentile* acceleration and perform **variance reduction using Z-scores**. (See Appendix G.)

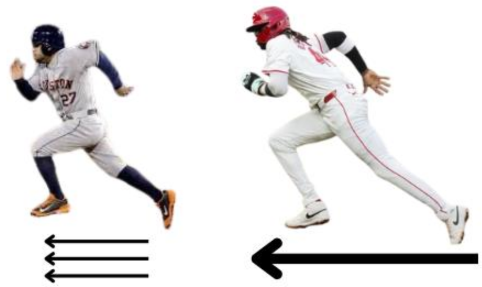

Figure 9: The runner on the left (Jose Altuve) is shorter and stockier, which typically allows for faster initial acceleration and explosive short bursts. The runner on the right (Elly De La Cruz), taller and lankier, may take longer to reach speed up but can achieve a higher top speed.

We define `start_running` as the first *timestamp of the three consecutive timestamps of the runner moving towards 1st base at 3+ ft/s.*

We set the **cutoff** at 250ms after his median `start_running` time, or 350ms after league median `start_running` time. If the player does not start running until then, we say the cutoff is "when he *should have* started running".

Reaching True Capacity through Effort

For estimating function ⊗ we consider two sports science concepts related to *force production*.

The first is the **non-linear relationship between effort and force in a stretch-shortening cycle (SSC) setting**. (See Appendix H.)

In general, a person's *perceived effort* is roughly equal to the *actual force* produced.[25] [26] However, if the movement involves a *stretch-shortening cycle*, such as jumping, *perceived effort* tends to be **much lower** than *actual force*.[27]

Based on data from Tokutake et al. (2024) and Hasabe et al. (2023) (see Appendix H.) we fit a **polynomial equation** to model the relationship between effort level and force production in a *short-distance sprint setting*. (Figure 10)

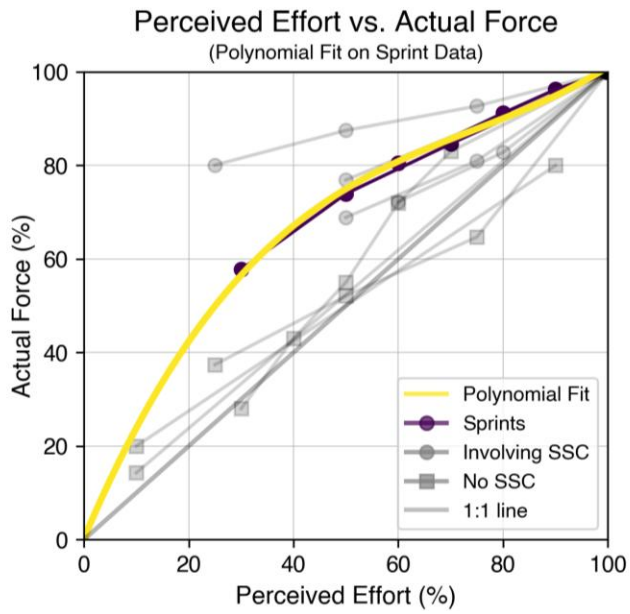

Figure 10: At 50% effort, a runner is able to produce nearly 75% of his maximum acceleration.

The second sport science concept is the **force-velocity relationship**.

The muscle's capacity to produce force **decreases** with increasing *muscle contraction velocity*, eventually reaching zero at some maximum velocity. (Figure 11)

[24] It is not entirely necessary that we have the "maximum" acceleration; we are primarily looking to specify the runner signature, which requires a set of "uncorrelated variables" that determine the runner's behaviors.
[25] West et al. (2005)
[26] Lauzière et al. (2012)
[27] Salles et al. (2010)

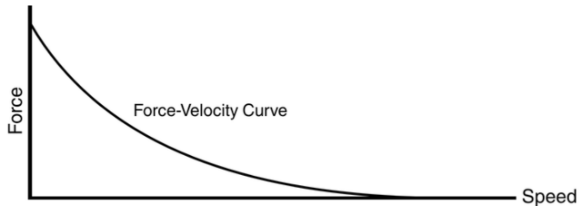

Figure 11: Standard force-velocity curve. As the muscle contracts faster, its capacity to produce force decreases.

In other words, **acceleration capacity is speed-specific**. Thus, for a given runner signature (true capacity), there is a specific maximum acceleration capacity at each speed.

To specify that, we employ (GBDT) `CatBoost` fitted on *runner signatures* with (instantaneous) *running speed.* The method is similar to Predicting `catch_timestamp`.[28] Furthermore, **pinball loss** objective is used for **quantile regression**[29]. (Figure 12) (See Appendix I.)

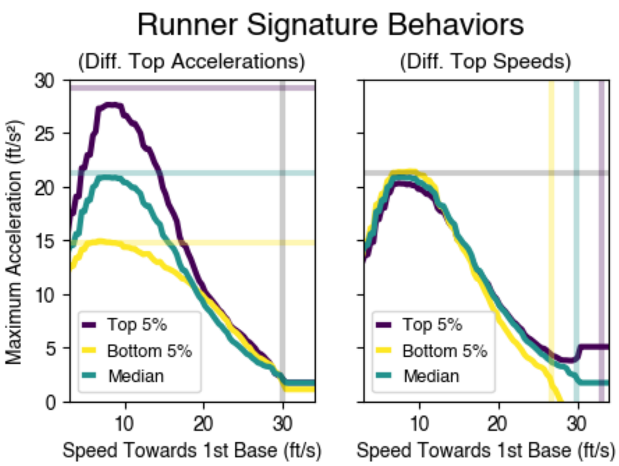

Figure 12: Comparison between acceleration(force)-velocity curves of different runner signatures (true capacities).[30] Semi-transparent lines indicate estimated maximum acceleration (left) and speed (right).[31] Grey line indicates median.

Putting it Together

Using the **acceleration(force)-velocity model** (Figure 12), we can determine the *maximum acceleration* at each speed, while the **force-effort model** (Figure 10) provides the *percentage of maximum force* the runner reaches with effort.

In addition, consider that there is a **minimum force** required to sustain a running speed; if the runner makes 0% effort, and produces *zero* force while running, he should *slow down* (and eventually stop). In addition, minimum force should require more effort to achieve at higher speeds.[32]

We arbitrarily set the *range of accelerations* by the players' *maximum acceleration.*[33] (Figure 13)

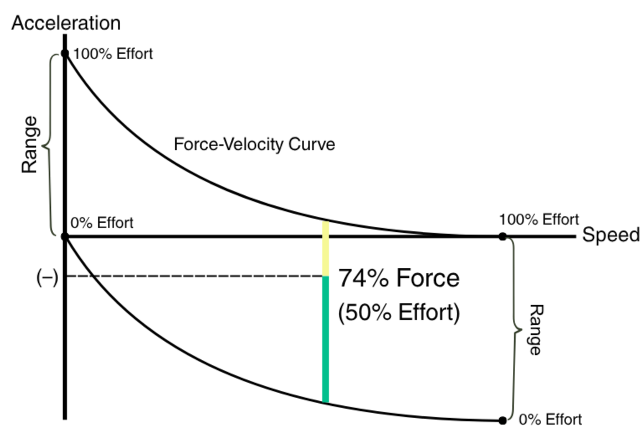

Figure 13: Visualization of function ⊗, i.e. relationship between effort level and acceleration. At high speeds, even >50% effort results in negative acceleration.

We also apply a 5% *lenience* to the maximum acceleration capacity[34], to accommodate the variance in acceleration at each effort level.[35]

[28] Bootstrap aggregating is also used. (See Appendix E.)
[29] Since the objective is to find *near-maximum acceleration* at given speed, not *mean acceleration*.
[30] Initial curves are directed upwards which is unexpected; We suspect this to be due to noise in the beginning of the run.
[31] 90th percentile was used as the prediction values matched input maximum speeds/accelerations.
[32] E.g. 50% effort at 30 ft/s should slow down the runner more than 50% effort at 20 ft/s.
[33] Or precisely, 95th percentile acceleration, as described in the model.
[34] I.e. If the player reaches 95% acceleration, it is considered 100% effort.
[35] Even if the player gives 100%, he will still underperform his "mean 100% acceleration" 50% of the time (under symmetrical distribution).

2.4 Hustle+

Hustle+ is calculated using the league average effort level (Figure 14) and a player's effort level:

$$Hustle+ = \frac{mean(player\ effort)}{mean(league\ effort)} \times 100$$

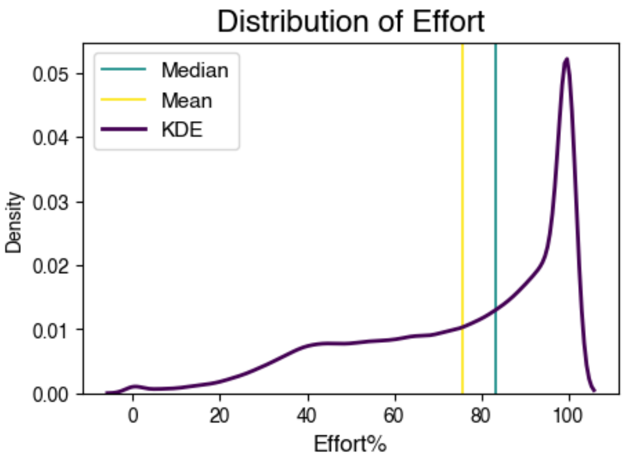

Figure 14: The distribution of effort is heavily left-skewed, with a large proportion stacked around 100%, and the remainder distributed between 0-80%. This suggests that players may be different modes: high effort and low effort.

2.5 Safe Probability Added (SPA)

We now pose the following questions:

"If he gave it everything, would he be safe?"

"Did hustling make a difference?"

We answer them by **simulating the run attempts at different effort levels** by integrating all the models described above.

Simulating Run Attempts

For each `play_id`, we start our *imaginary batter* at the `start_running` time (or the `cutoff` time).

Then, on every 50ms interval, we **predict** the next interval's *distance/speed/acceleration* towards 1st base using the estimated true capacities, and **median effort level.**[36] We continue this process until the expected `catch_timestamp`[37], at which point we get *simulated runner* `distance_from_1st`. (This is where the batter "would have" been if he gave "average" effort.)

We also simulate run attempts at 100% effort. (This gives us where the batter "would have" been if he gave "everything".)

Using the batter position, we calculate Pr(`Safe`) using the Predicting Outcomes model.

Calculating Safe Probability Added

For each run, we have:

1. Pr(`Safe` if "average" (league) effort)
2. Pr(`Safe` if "actual" effort)

Safe Probability Added (SPA) is the differences between the two probabilities, summed over the player's entire season:

$$SPA = \sum_{All\ runs} [\Pr(\text{Safe if "average" effort}) - \Pr(\text{Safe if "actual" effort})]$$

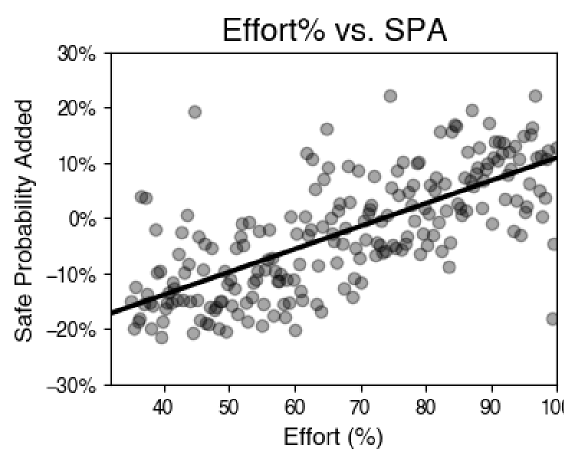

Figure 14: Linear relationship between Effort% and SPA. A 10% additional effort leads to a 5% higher safe probability.

[36] We use *median* effort level as we believe it best represents the "typical" run.

[37] And also at the timestamp when the ball intersects target plane.

ANALYSIS

3.1 Can We Quantitatively Measure Hustle?

Yes.

We see that our measure of hustle is normally distributed, with minimal skew. (Figure 15)

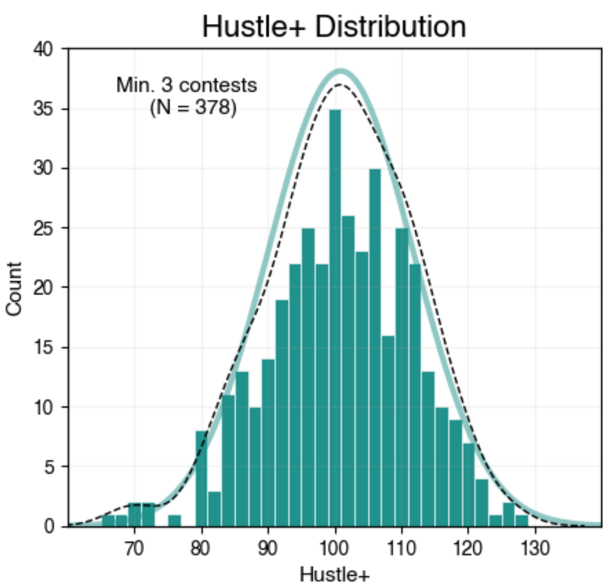

*Figure 15: There is a considerable amount of variance in Hustle+; some players exert up to **30%** more effort on average. We expect the variance to decrease with more (than 3) contests per player (e.g. full-season data).*

Furthermore, **Hustle+ is agnostic of talent.** (Figure 16)

Interestingly, the distribution of Safe Probability Added trends leftwards with more contests. (Figure 17)

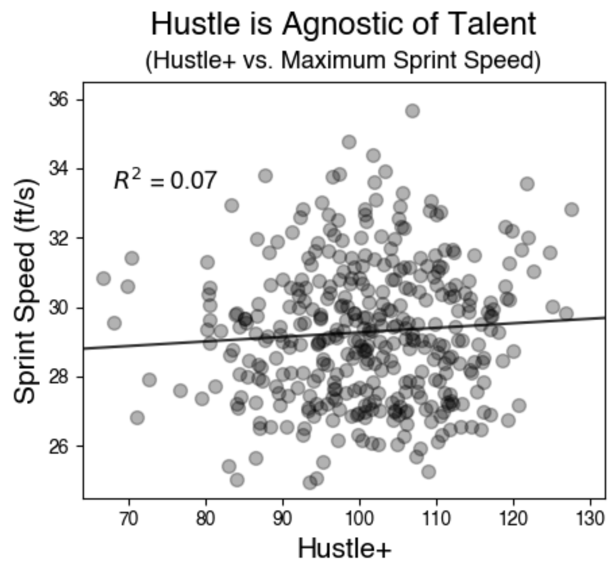

*Figure 16: There is little to no correlation between Hustle+ and sprint speed, which suggests that Hustle+ is an **independently** informative metric.*

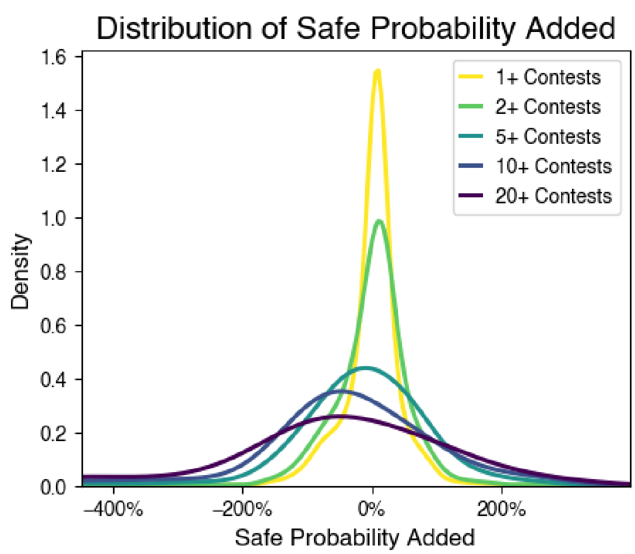

Figure 17: (+) SPA implies that the player "added" safe probability (compared to his talent-based expectation). Interestingly, the SPA distribution trends leftwards with more contests, which suggests that it is much easier to "miss out" on safe opportunities by hustling less, than it is to convert an out into a safe by hustling more.[38]

[38] The "average" (median) run is a high effort run. So, it is much easier to hustle less than average.

3.2 Is There Such a Thing as a "Hustle Culture"?

Team-level hustle shows significant discrepancies. (Table 1)

Team	Hustle+	Effort%	Contests
RZQ	107.0	80.8	743
FBP	106.7	80.6	40
DYE	104.8	79.2	123
QEA	102.8	77.7	276
YJD	99.1	74.9	744
OXG	95.1	71.8	62

Table 1: Team-level Hustle+ and Effort% (Min. 40 contests) Just within the 6 teams, there is nearly a 10+% effort gap between the highest and lowest effort teams.

3.3 Can Teams Actually Win Games by Hustling?

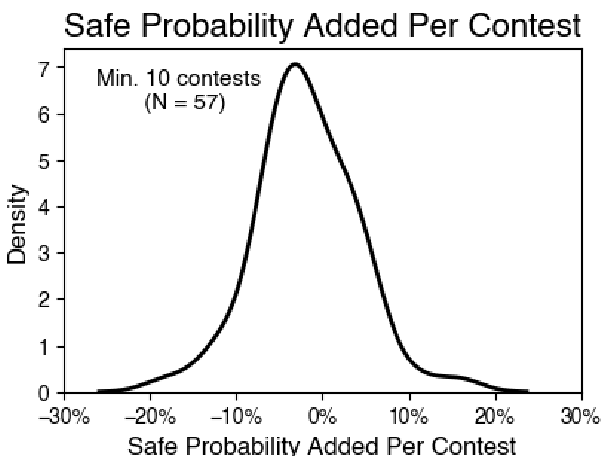

*Figure 19: Similar to Hustle+, there is a considerable amount of variance; some players add up to **20%** chance of safe on average. Like Hustle+, we expect the distribution to narrow with more contests (making the upper end closer to 10%).*

The average baseball game has 11.2 contests at 1st base[39].

Thus, **if every player hustled an additional 10% to add 5% to their safe probabilities** (Figure 14), **across the 162-game season, the team would add 90.7 more runners safe at 1st base.** (Figure 19)

This is because baseball is a *team sport*. For the majority of *individual* players, that amounts to roughly 5 hits[40] a season (worth ~0.010 on-base percentage).

Anyone can hustle, and the team benefits if everyone does.

[39] The average game in our dataset has 11.21 contests.

[40] Recall the fact that many safes at 1st are due to errors, or fielder's choice (not hits).

CONCLUSION

4.1 Conclusion

We have created two metrics, **Hustle+** and **SPA**, which quantify a player's hustle within contests at 1st base.

4.2 Future Work

While the initial results of **Hustle+** and **SPA** were promising, we failed to address numerous assumptions, such as:

1. No player is injured
2. The average effort levels throughout the run[41] are constant
3. Players who have recorded elite speeds have recorded their maximum speeds
4. Only the 1st baseman receives the ball at 1st
5. The batter does not influence the throw to 1st
6. Effort levels are entirely determined by the person's intentions
7. No promotion and relegation of players amongst minor league levels

Furthermore, there are various areas for improvement such as:

1. Integrating fatigue into the capacity to produce force
2. Creating a balanced, yet unbiased catch outcome predictions[42]
3. Calculating hustle for other running events (e.g. running around the bases)
4. Calculating hustle for non-running events (e.g. diving for the ball)
5. Quantifying the coach's ability to instill "hustle culture" into a team

[41] The earlier parts of the run have much greater leverage for arriving at 1st base quickly.
[42] See Appendix D.

ACKNOWLEDGEMENT

This project was made possible through the University of Toronto Sports Analytics Club (**UTSPAN**).

Thank you to Billy Fryer and Dr. Meredith Wills for the support throughout the entirety of the project.

Thank you to David Awosoga for providing the code for GIF animations of baseball plays.

Thank you to Louis Ryan Tan for providing guidance on the machine learning implementations.

Thank you to everyone who helped proofread the paper.

APPENDIX

A. Inferring Throwing Target

Camera-based tracking system cannot detect the ball when acquired by a player (occlusion). Furthermore, it does not capture the player's body language (only center of mass). Thus, we have the following difficulties:

1. If an infielder ends up *not* throwing the ball, we cannot know if he intended to. (E.g. bobbled ball)
2. If an infielder completely misses his target, we cannot know what the original target was. (E.g. throw to 1st base ending up at 2nd base)

To avoid these issues, we make the following assumptions:

1. Bobbles never happen
2. Infielders always throw towards a particular base[43]
3. Infielders generally throw the ball near the target[44]

Following the assumptions, we use two mechanisms for inferring throw target:

1. Determine which base closed the longest (logged)[45] distance during the initial[46]
2. If the two most closed distances were similar[47], determine which base was closest at the end of the trajectory

The first mechanism is necessary, as it accounts for overthrown balls (which the second mechanism does not account for).

[43] Or at least, far from other bases.
[44] Same as 43.
[45] Logging increases the distance factor near the bases. This is used to account for short-distance throws, where target is often slightly off the base. (E.g. underhand toss to 2nd base)
[46] 200ms interval starting from ball release
[47] Within ratio of 3.0

B. Cubic Smoothing Splines

We utilize the `SciPy Interpolate` package[48] and set `k=3` (for cubic smoothing).

Smoothing factor was calculated individually for every run.

(See analysis code for additional details.)

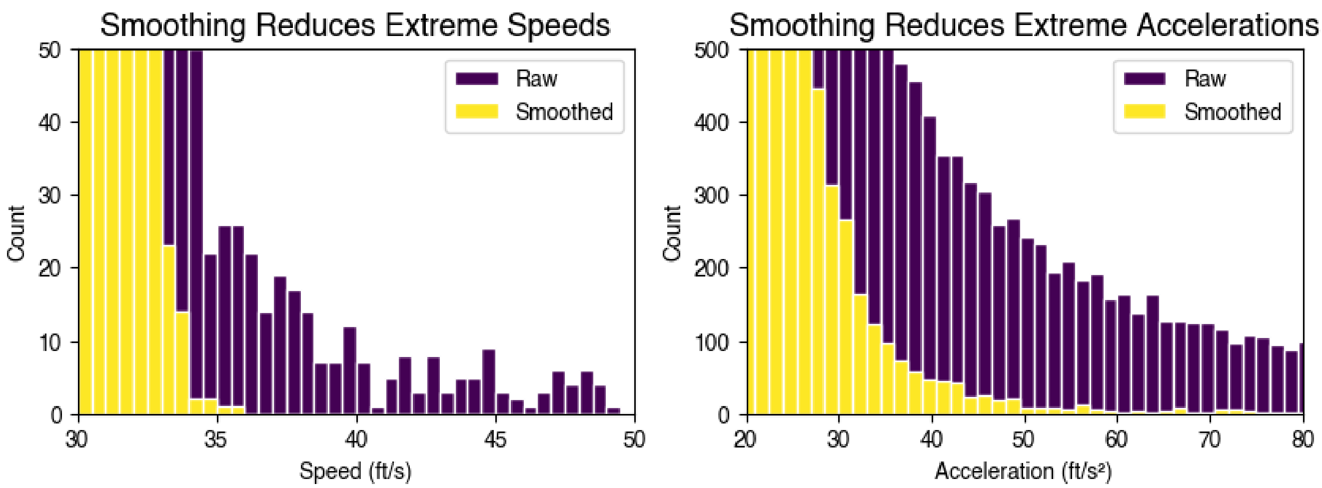

Figure 20: The right-tail of the speed and acceleration distributions. Smoothing nearly eliminates all the extreme values.

It may appear that the *smoothed* datapoints still contain a large number of outliers, since the highest recorded speeds in the Statcast system are 31-32 ft/s. This is due to the fact that the **Statcast system measures sprint speeds across 1-second intervals**; baseball players rarely have the opportunity to comfortably maintain their maximum speed for that period of time.

We compare our calculated sprint speeds with the "Statcast versions" using a 1-second intervals. (Figure 21)

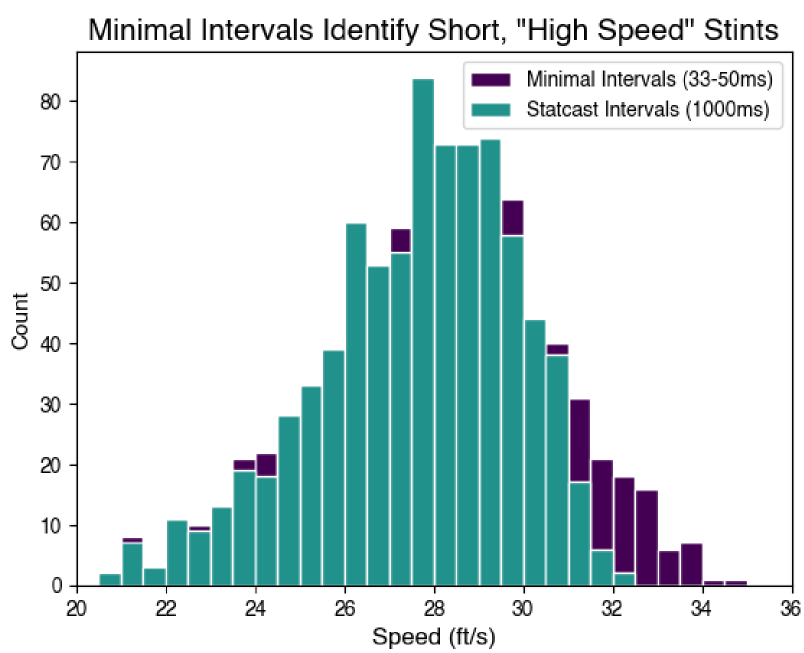

Figure 21: The speed measurements using minimal intervals include many more instances greater than 32 ft/s, which rarely occurs longer than 1 second.

[48] https://docs.scipy.org/doc/scipy/reference/interpolate.html

C. Classifying Catch Outcomes

Using ball/player tracking data, we can identify whether (and where) the ball was deflected[49], or acquired (by a fielder). Thus, we can firstly classify whether an infield throw to 1st base was `Caught`/`UnCaught` by the 1st baseman.

If the ball was `UnCaught`, i.e., deflected at some point without being acquired, then it is classified as `Dropped`/`Passed` depending on where the ball *ended up*:

1. If it remained within 10 feet[50] from 1st base — `Dropped`
2. Otherwise — `Passed`

If the ball was `Caught`, then it is classified as `CatchOn`/`CatchOff`. However, the position data alone makes it difficult to ascertain if the 1st baseman was *touching the base*. To solve this problem, we first restrict the catches to "definitely" `CatchOn`[51], and fit a **logistic regression model** on it to calculate the probability of `Safe` given the batter's distance[52] from 1st base at `catch_time`.

Based on the batter's distance at `catch_time`, we have:

Pr(`CatchOn`) = 1 - Pr(`Safe`)

In other words, (for "not definitely" `CatchOn` throws) if the probability of `Safe` was high (say, 90%) and the outcome was `Out`, we attribute a low (10%) probability that the 1st baseman caught the ball on the base.

We manually visit the 18 cases where a `Passed`[53] ball was labeled with an `out`. 10 of such cases were due to the batter *overrunning*; the remainder were mislabeled.

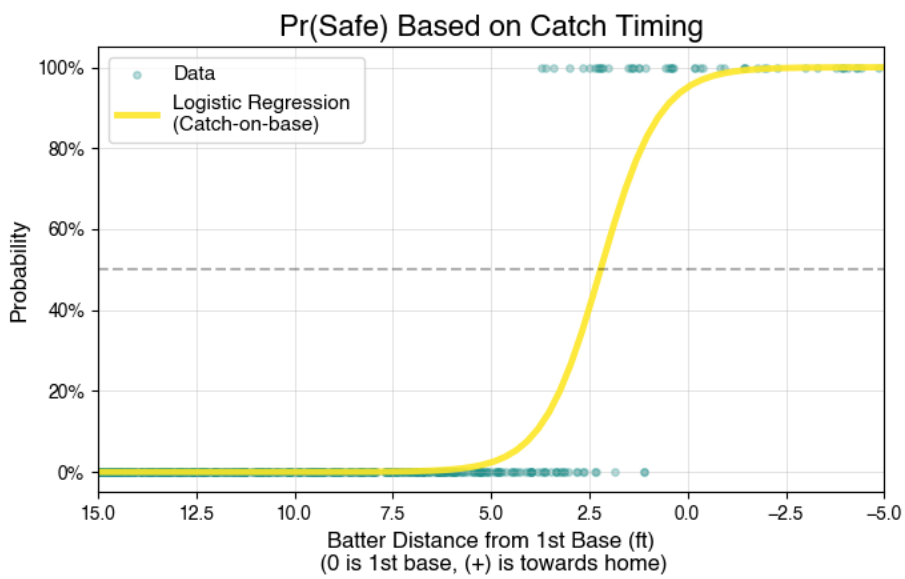

Figure 22: Safe probability stays near zero until around 5ft, where each feet significantly increases the probability.

[49] By any object, such as players or fences.
[50] This is the "catchable region" defined in Predicting Catch Outcomes
[51] Throws where the catcher was within 3 feet from 1st base, and in the infield circle, and the ball was caught within 6 feet from 1st base, and also in the infield arc.
[52] See Denoising Position Data to see how distance was calculated.
[53] Recall that Passed is the only case where the batter has the chance of overrunning; Other catch outcomes had zero cases.

D. GBDT Fine-tuning + Ensembling

Using the following features:

1. The orientation of the target plane;
2. The location of the ball's on the target plane;[54]
3. The ball's travel direction (when ball intersects target plane);
4. The batter's distance from 1st base (when ball intersects target plane).

Each model is trained with 10,000 rounds (early stopping at 1,000) with *log loss* objective with balanced weights.[55] Hyperparameters are tuned using `Optuna` across 10 iterations.

Each of the best iterations of the models are combined into an ensemble via *linear combination*. Similar to above, *log loss* objective is used on test data to optimize the weights.[56]

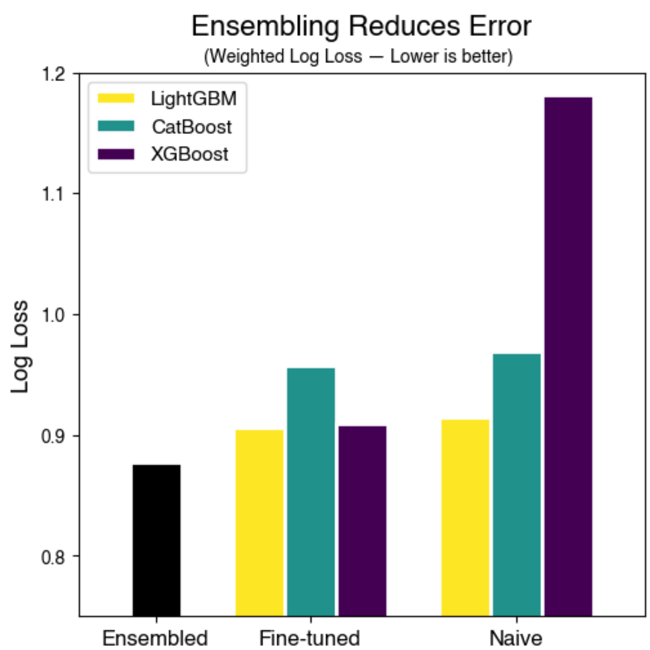

Figure 23: Each GBDT model has a different inductive bias. By averaging the predictions, we get a more holistic prediction which outperforms any single model.

(See analysis code for additional details.)

[54] The y coordinate on the target plane was later replaced with absolute ball height, as it produced better performance.
[55] I.e. Smaller classes were weighed more to offset class imbalances. This is a major reason why we do not compare the predicted results with the actual results (the smaller classes become overrepresented). Instead, we strictly limit our comparisons between simulations at different effort levels.
[56] For the linear combination.

E. Bootstrap Aggregation

The expected pattern for predicting catch timing is rather **simple**:

> *The further the ball lands from the target, the later it is expected to be caught.*[57]

Hence, we are more concerned with **preventing overfitting** rather than finding the optimal tree.

Hyperparameters are "tuned", but main objective of the tuning process is to create multiple models with varying *inductive biases* for the purpose of **variance reduction via bootstrap aggregating**.

Unlike the methods described in Appendix D., ensembling across three GBDT models is not used; the `CatBoost` framework, alone, produces similar results.

Each model is trained with 1000 rounds (early stopping at 10) with a *RMSE* objective. `Optuna` is used to create 100 hyperparameter combinations. **All 100 iterations**[58] of the models are included in the bootstrapping stack.

	On Train Data		On Test Data	
Result	Naïve	Tuning + Bagging	Naïve	Tuning + Bagging
RMSE	559.3	696.6	921.5	816.4
R^2	0.74	0.67	0.39	0.46

Table 2: Bagging produced superior performance on test data, suggesting more consistent predictions.

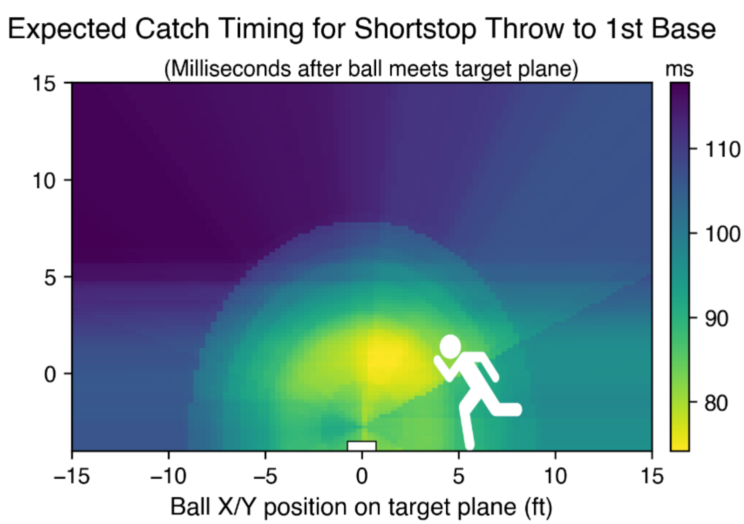

Figure 24: When the ball lands near the center of the target plane (0,0), the 1st baseman is expected to catch the ball earlier. Just by throwing more accurately, the fielder can save up to 40ms (worth 1.2ft).[59]

(See analysis code for additional details.)

[57] As the 1st baseman cannot reach towards the ball but rather move sideways/upwards to retrieve it.

[58] As the "tuning process" was not intended to tune the hyperparameters.

[59] Assuming running speed at 30 ft/s

F. Softened MAP Estimation

To predict *intrinsic* maximum speed in the "unseen" ranges, we instantiate a **prior** in our prediction.

As a *prior*, we use the distribution of maximum speeds for batters with numerous contests in the database. We use the highest cutoff for number of contests, without significantly violating the Gaussian shape, which is 5 contests.

The limitation of this naïve **MAP estimation** approach is that the **closed-form formula only considers *contest counts*[60] and not the *maximum recorded speed*** (though used in the derivation of it). This means that, two players, both with three runs, with wildly different recorded speeds, have the same MAP estimation.[61]

Thus, we "soften" the prediction to be the **maximum** between the MAP estimation and the maximum recorded speed.

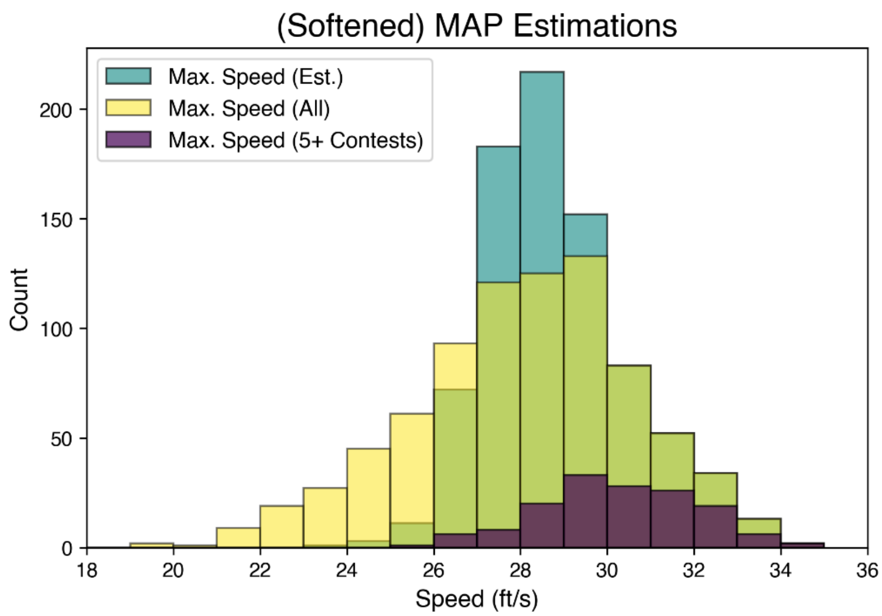

Figure 25: Distribution of recorded maximum speeds and the (softened) MAP estimations. The right-end of the distributions are equal; only the players with recorded maximum speeds below the mean maximum speed in the prior have an updated estimated maximum speed.

[60] See the proof section for the closed-form formula

[61] The implication is that MAP estimation may be lower than recorded speeds for players on the upper half of the distribution.

Proof

$$
\begin{aligned}
Y_i = \max\{Y_{i1}, Y_{i2}, \ldots, Y_{in_i}\} &= \theta_i - \min\{E_{i1}, E_{i2}, \ldots, E_{in_i}\} \\
&= \theta_i - \min_j E_{ij}
\end{aligned}
$$

$$\text{where } E_{ij} = \theta_i - Y_{ij} > 0 \text{ by assumption.}$$

Assume $\forall E_{ij},\ E_{ij} \overset{\text{iid}}{\sim} \text{Exp}(\lambda)$.

$$
\begin{aligned}
\text{So, } F_{E_{ij}}(e_ij) = \lambda e^{-\lambda e_ij} \\
\geq 0
\end{aligned}
$$

So, $\min_j E_{ij} \sim \text{Exp}(n_i\lambda)$.

$$\text{Let } \min_j E_{ij} = M_i.$$

$$\text{So } F_{M_i}(m) = n_i\lambda e^{-n_i\lambda m_i}, \qquad m \geq 0.$$

With $Y_i = \theta_i - M_i$:

$$\text{Let } h(M_i) = \theta_i - M_i \quad (\text{hence } Y_i = h(M_i)).$$

Since M_i is a continuous r.v. and $h(M_i)$ is differentiable and strictly decreasing, Y_i is continuous and:

$$
\begin{aligned}
f_{Y_i}(y_i) &= \left| \frac{d}{dy_i} h(y_i) \right| f_h Y_i(h(y_i)) \\
&= \left| \frac{d}{dy_i} h(y_i) \right| f_{\theta_i - y_i}(\theta_i - y_i) \\
&= \left| \frac{d}{dy_i} h(y_i) \right| f_{M_i}(m_i) \\
&= (1)\, n_i\lambda e^{-n_i\lambda(\theta_i - y_i)} \\
&= p(y_i \mid \theta_i, n_i, \lambda)
\end{aligned}
$$

Given:

$$p(y_i \mid \theta_i, n_i, \lambda) = n_i\lambda e^{-n_i\lambda(\theta_i - y_i)}.$$

$$p(\theta_i \mid y_i) \propto p(y_i \mid \theta_i; n_i, \lambda)\, p(\theta_i).$$

Assume:

$$p(\theta_i) \sim \mathcal{N}(\mu, \sigma^2).$$

Then likelihood function:

$$L(\theta_i) = n_i\lambda e^{-n_i\lambda(\theta_i - y_i)} \frac{1}{\sigma - \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{\theta_i - \mu}{\sigma} \right)^2 \right).$$

Proof (cont.)

Log-likelihood:

$$\ell(\theta_i) = \log L(\theta_i) = -n_i\lambda(\theta_i - y_i) - \frac{1}{2}\left(\frac{\theta_i - \mu}{\sigma}\right)^2 \text{ Ignoring Constants.}$$

Differentiate:

$$\ell'(\theta_i) = -n_i\lambda - \frac{\theta_i - \mu}{\sigma^2}.$$

Set derivative to zero:

$$-n_i\lambda - \frac{\theta_i - \mu}{\sigma^2} = 0 \quad \Rightarrow \quad \hat{\theta}_i = \mu - n_i\lambda\sigma^2.$$

Since $\theta_i \geq Y_i$:

$$\hat{\theta}_i^{(MAP)} = \max\{Y_i, \mu - n_i\lambda\sigma^2\}$$

We can estimate λ using method of moments.
Since:

$$E[Y_i \mid \theta_i] = \theta_i - \frac{1}{n_i\lambda},$$

$$\bar{Y} \approx \frac{1}{N}\sum_{i=1}^{N}\left(\theta_i - \frac{1}{n_i\lambda}\right).$$

$$\bar{Y} \approx \mu - \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i\lambda}.$$

So:

$$\lambda \approx \frac{1}{\mu - \bar{Y}} \cdot \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i} = \hat{\lambda}.$$

G. Variance Reduction using Z-scores

Unlike maximum speeds, the *intrinsic* maximum accelerations remain within the “seen” ranges. (Table 3)

	All Batters	50+ Contests
Sample Size (N)	824	261
Mean (μ)	21.21	21.19
Variance (σ^2)	75.10	24.47

Table 3: Means and variances of maximum acceleration distribution.

Thus, it is not necessary to perform MAP estimation; with a Gaussian assumption, we simply **translate the measured 95th percentile acceleration to a lower-variance distribution using Z-scores.** (Figure 26)

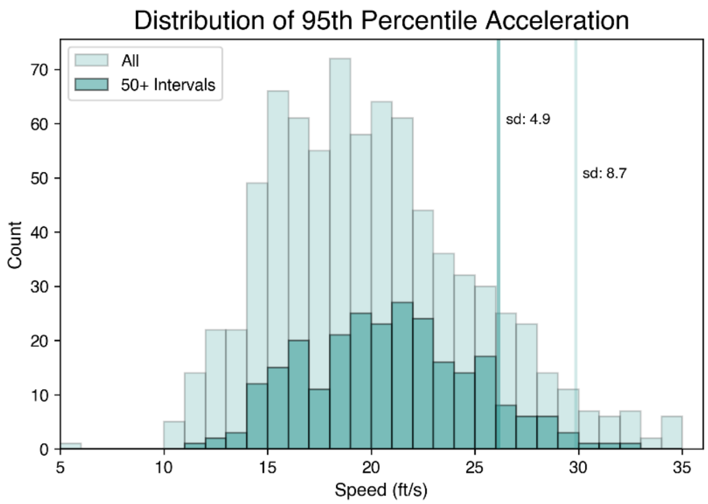

Figure 26: The maximum accelerations (All) are mapped to the narrower distribution (50+ Intervals). See the vertical lines; an acceleration 8.7 sd (standard deviations) above mean is mapped to 4.9 sd above mean.

H. Sport Science Literature

We find **10 studies** that measures *force production* in relation to *perceived effort.* (Figure 27) (Table 4)

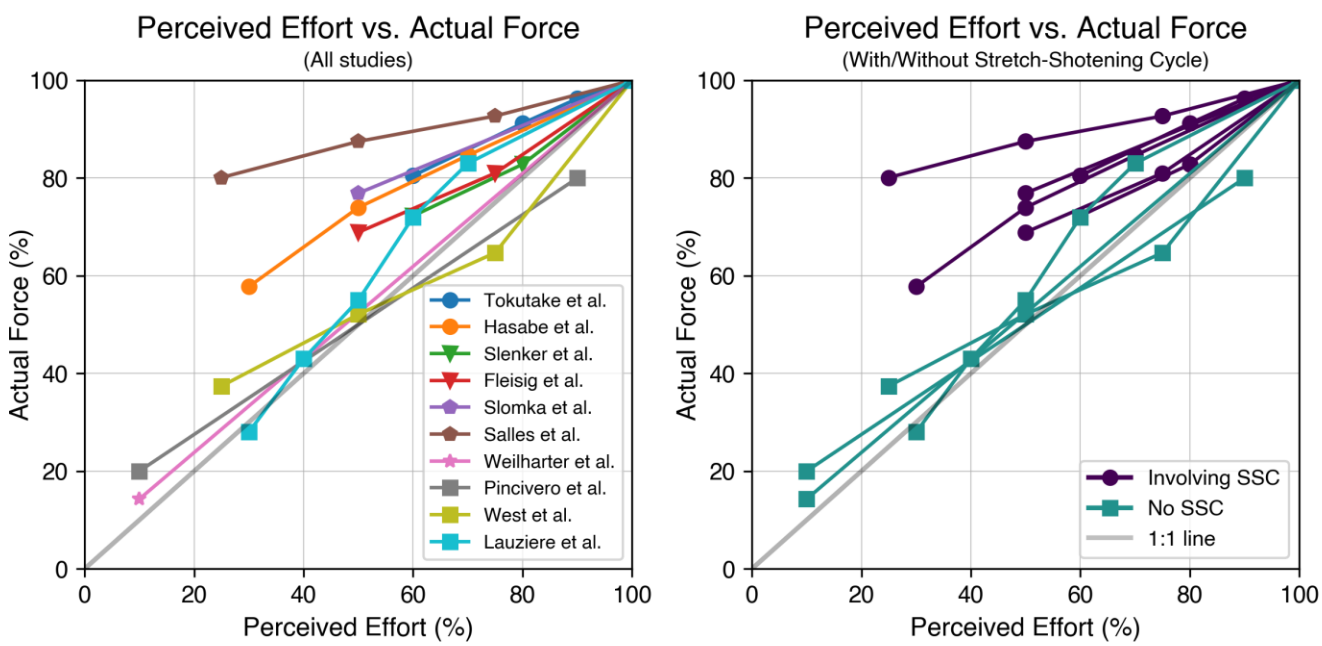

Figure 27: When the event does not involve stretch-shortening cycle (SSC), there is a 1-to-1 relationship between perceived effort and actual force; when it does not, actual force tends to be much higher than effort.

Study	Sample Size (N)	Activity	Measure	SSC	Absolute/Perceived Intensity[62]
Tokutake et al. (2024)	18	Sprint	Peak Speed (m/s)[63]	T	89.72% / 60% 95.50% / 80% 98.11% / 90%
Hasabe et al. (2023)	14	Sprint	Peak Speed (m/s)	T	76% / 30% 85% / 50% 92% / 70%
Slenker et al. (2014)	29	Throw	Peak Speed (m/s)	T	85% / 60% 91% / 80%
Fleisig et al. (1996)	27	Throw	Peak Speed (m/s)	T	83% / 50% 90% / 75%
Słomka et al. (2009)	20	Jump	Total Work (W)[64]	T	76.9% / 50%
Salles et al. (2010)	10	Jump	Peak GRF (F)	T	80.1% / 25% 87.5% / 50% 92.7% / 75%
Weilharter et al. (2024)	26	Handgrip	MVIC (F)	F	Linear from 10% to 70% / 1 to 10 RPE[65] [66]
Pincivero et al. (2003)	30	Knee Extension	MVIC (F)	F	Linear from 20% to 80% / 1 to 9 RPE
West et al. (2005)	30	Knee Extension	MVIC (F)	F	37.4% / 25% 52.1% / 50% 64.7% / 75%
Lauzière et al. (2012)	14	Knee Extension	MVIC (F)	F	29% / 30% 43% / 40% 55% / 50% 72% / 60% 83% / 70%

Table 4: Organized table of studies measuring the relationship between absolute and perceived intensity.

I. Quantile Regression

Pinball Loss:

$$L_\tau(y, \hat{y}) = \begin{cases} \tau\,(y - \hat{y}), & \text{if } y \geq \hat{y} \\ (\tau - 1)\,(y - \hat{y}), & \text{if } y < \hat{y} \end{cases}$$

[62] We uniformly interpret *absolute relative intensity* as *absolute relative force*.

[63] Since force(F) is proportional to acceleration(a) (F=ma), relative speed is equivalent to relative force. Acceleration(a) is proportional to squared velocity(v^2) ($2\Delta da = v_f^2 - v_i^2$), so we can estimate relative acceleration by taking squared velocity.

[64] Force(F) is also proportional to work(W) (W=FΔd)

[65] RPE's were interpreted as percentages (i.e. 9 RPE as 90%, etc.)

[66] Since at maximum effort, absolute relative intensity was 70%, all values were scaled by (100%/70%)

REFERENCES

Americal Sports Medicine Institute. (1996). Kinematic and kinetic comparison of full effort and partial effort baseball pitching. *Proceedings of the 20th Annual Meeting of the American Society of Biomechanics. Atlanta, GA, United States.*, 151–152. https://www.asbweb.org/wp-content/uploads/1996ASBmeetingproceedings.pdf

Hasebe, F., Miura, M., Ono, K., Goto, Y., Nakata, H., Ohtaka, C., & Fujiwara, M. (2023). Effects of subjective effort on overground and treadmill running: a comparative analysis. *The Journal of Physical Fitness and Sports Medicine*, *12*(2), 45–57. https://doi.org/10.7600/jpfsm.12.45

Lauzière, S., Dubois, B., Brière, A., & Nadeau, S. (2011). Magnitude of force perception errors during static contractions of the knee extensors in healthy young and elderly individuals. *Attention Perception & Psychophysics*, *74*(1), 216–224. https://doi.org/10.3758/s13414-011-0223-6

Pincivero, D. M., Dixon, P. T., & Coelho, A. J. (2003). Knee extensor torque, work, and EMG during subjectively graded dynamic contractions. *Muscle & Nerve*, *28*(1), 54–61. https://doi.org/10.1002/mus.10393

Salles, A. S., Baltzopoulos, V., & Rittweger, J. (2010). Differential effects of countermovement magnitude and volitional effort on vertical jumping. *European Journal of Applied Physiology*, *111*(3), 441–448. https://doi.org/10.1007/s00421-010-1665-6

Slenker, N. R., Limpisvasti, O., Mohr, K., Aguinaldo, A., & ElAttrache, N. S. (2014). Biomechanical comparison of the interval throwing program and baseball pitching. *The American Journal of Sports Medicine*, *42*(5), 1226–1232. https://doi.org/10.1177/0363546514526152

Słomka, K. J., Jaric, S., Sobota, G., Litkowycz, R., Skowronek, T., Rzepko, M., & Juras, G. (2018). Effects of reduced effort on mechanical output obtained from maximum vertical jumps. *Motor Control*, *23*(2), 205–215. https://doi.org/10.1123/mc.2017-0086

Taber, C., Bellon, C., Abbott, H., & Bingham, G. E. (2016). Roles of maximal strength and rate of force development in maximizing muscular power. *Strength and Conditioning Journal*, *38*(1), 71–78. https://doi.org/10.1519/ssc.0000000000000193

Tokutake, G., & Ueta, K. (2024). Cross sectional study about Intra-Individual and Inter-Individual variability of subjective effort in submaximal sprint running in collegiate male sprinters. *Asian Journal of Sports Medicine*, *15*(3). https://doi.org/10.5812/asjsm-147182

Weilharter, F., Rewitz, K., Halperin, I., & Wolff, W. (2024). The relationship between prescribed ratings of perceived exertion and force production in repeated isometric contractions. *Psychology of Sport and Exercise*, *73*, 102657. https://doi.org/10.1016/j.psychsport.2024.102657

West, S. J., Smith, L., Lambert, E. V., Noakes, T. D., & St Clair Gibson, A. (2005). Submaximal force production during perceptually guided isometric exercise. *European Journal of Applied Physiology*, *95*(5–6), 537–542. https://doi.org/10.1007/s00421-005-0004-9