It’s BRI, Bro:
Evaluating **B**ase**R**unning **I**ntelligence

Team 199 - Undergraduate Division

Abstract:

In my analysis, I aim to obtain a way of evaluating a player's and team's baserunning choices in critical and fast-paced situations. I first get two probabilities. One is the breakeven probability of a runner advancing that tells the runner what the *needed* probability of being safe is to make it worth attempting to advance (calculated using a run expectancy matrix), and the other is the *actual* probability of a baserunner being safe on a given play (calculated using the model I built). I then compare these two probabilities to see if the runner made the right decision, to stay or advance. This adds another dimension for baserunning critique as it enables analysts to observe aggressiveness and good decision making, regardless if the runner was safe or not.

Table of Contents

Introduction

This summer, my dad, brothers, and I have been spending more time and money than we probably should on used, beat up gear. One of the drills we like to do is trying to field ground balls with an imaginary base-out state. So when there is a situation with a runner on third or second with less than 2 outs, my dad would yell "Hold the runner! Hold the runner!" even on plays where the ball took the nastiest hops or the ball would go right through my glove and then there would no longer be an imaginary runner to hold because he already advanced! I then started to wonder, "On infield groundballs, at what point should a runner heed the hold of a fielder and not try to advance?" This is a very crucial part of the game because this only happens with runners on second and/or third with less than two outs which are golden opportunities to score. This means the difference between baserunners who can find that point of inflection and those who can't is adding to the Win or Loss column.

From the research that I found when analyzing baserunning decisions, the only probability observed is the probability needed to be safe in order to make it worth the risk to advance which is calculated using run expectancy values (Appendix F).

The math used is:

$$\text{RE Safe}(p)+\text{RE Out}(1-p) = \text{RE Stay}$$

$$p = \frac{\text{RE Stay} - \text{RE Out}}{\text{RE Safe} - \text{RE Out}}$$

where p is the probability of the advancing runner being safe, and Safe, Out, and Stay refer to the examined baserunner's possible outcomes.

Society for American Baseball Research (SABR) (Reference #1) wrote an article to provide mathematical evidence why a runner should never make the first or last out at 3rd base, and they used the threshold probabilities to show that the probability required to be safe at 3rd with 1 out is less than the probability required to be safe at 3rd with 0 or 2 outs. Similarly, a Detroit Tigers community blog, Bless You Boys (Reference #2), used the same math but in a context much closer to the scenarios I analyzed. The writer was critiquing the Tiger's seemingly aggressive strategy of using the contact play in almost every situation with men on 1st and 3rd and less than 2 outs. However, the writer came to realize that the probability needed to be safe was actually much lower than one might have thought. More explicitly, when the writer was discussing a particular play in which the aggressiveness seemed unreasonable, he stated, "we can find out he only needed to have a 3.7% chance of being safe to 'breakeven' and make the decision a good one."

Sure 3.7% seems like a low enough threshold to make it justified to try to advance, but what if the true probability of being safe is 3.5%? In anything, a value doesn't hold any weight unless it's compared in context. For example, having a .300 batting average is only impressive because most hitters hit somewhere in the .200s. By the same logic, the 3.7% threshold has no value unless we are able to compare it to the "true" probability of the runner being safe.

Therefore, my solution is to create a model that predicts the probability of the baserunner being safe on infield ground balls, taking in many of the play's characteristics such as where the ball was hit, exit velocity, etc. The idea is to take the probabilities received by the model and compare them to the threshold probabilities and to do this for each play. This will give us an accurate way to verify a baserunner's intentions and evaluate their decision making in high-leverage situations. Being able to understand this will provide teams with a new way to observe if their baserunning tendencies align with where they would like to be, and it also gives teams a new feature to look at when scouting, signing, or trading for players.

Data

Thanks to SportsMEDIA Technology for sharing their data for 250+ minor league baseball games, I had access to player and ball tracking data as well as event data which describes what happened in the play (e.g. HR, BIP) for all plays. The plays I analyzed were infield ground balls that only occurred when there were less than two outs and runners were in the following positions : 101, 011, 010, 001 (Appendix F.1). Because the data was so raw and didn't include the number of outs, had many missing values for what inning it was, nor was there official verification if a baserunner was safe on a play, most of my time spent on this project was cleaning and imputing the data to fill in missing values and predicting when an inning was over (Appendix B.1), how many outs there were (Appendix B.5), and if a runner was safe (Appendix C.2). I also should mention that the player tracking was almost completely reliable, but the player ID was missing for many of them, making it so the quality of individual team and player analyses suffered, but overall analysis remained strong.

Methodology

An important part of my analysis is that I am not trying to build a tool that will help the baserunner make better decisions in real time, but am instead creating a way to measure how good a baserunner's feel of the game is. A good baserunner won't know the exit velocity or the exact location where the fielder will receive the ball, but if he has good baserunning intelligence, he may be able to extract the same value based on what little he does see.

I then calculated exit velocity, arm strength, sprint speed, and several other metrics that could predict whether a baserunner was safe or not (Appendix C.1). I used an XGBoost model and achieved an accuracy score of 92% (Appendix D), however the data used to train the model is quite skewed (Appendix C.2). Regardless, the proof of concept is still strong and many of the model's predicted probabilities are certainly reasonable.

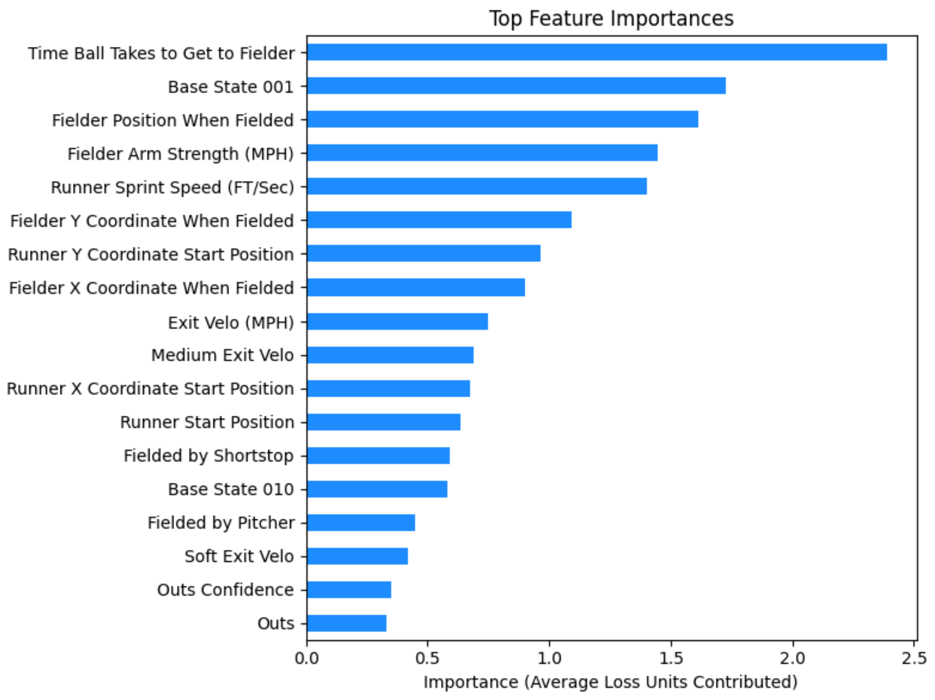
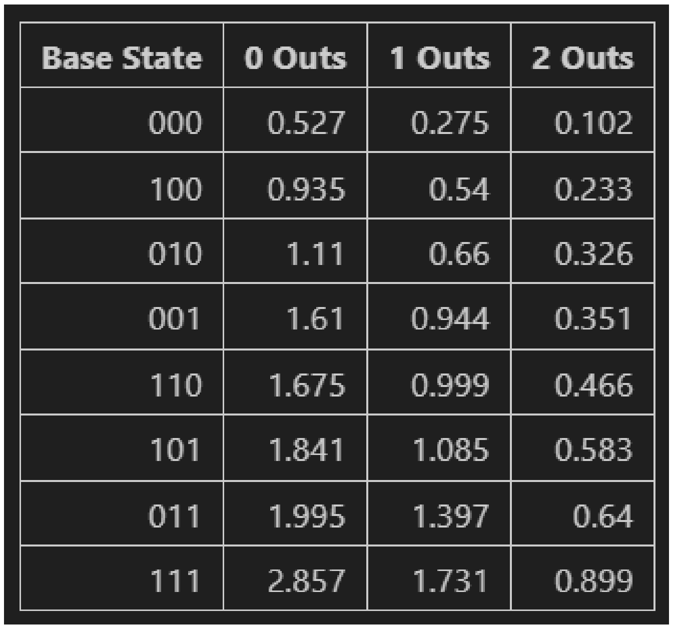


Figure 1: Bar chart showing which features are most important to the predictability of a baserunner's safety. It's not important to understand what the values mean but it is interesting to notice that the fielder's characteristics have more value than the runner's.

Another important assumption that I'm making is that each baserunner's safety probability is independent. This is mainly for the 011 and scenario where the 2B runner is only able to advance if the 3B runner advances. Further, this assumption is especially important when solving for the safety probability threshold because it requires the analyst to assume there are only a few possible outcomes in a given play (Appendix F.3).

After I built the model, I then constructed my own run expectancy matrix that can take in any date range and any minor league level and return an RE matrix.



Base State	0 Outs	1 Outs	2 Outs
000	0.527	0.275	0.102
100	0.935	0.54	0.233
010	1.11	0.66	0.326
001	1.61	0.944	0.351
110	1.675	0.999	0.466
101	1.841	1.085	0.583
011	1.995	1.397	0.64
111	2.857	1.731	0.899

Figure 2: Run expectancy matrix for two months worth of 2018 Low-A minor league baseball games taken from milb.com (Reference #3). Only two months were used because of runtime limits on my computer which explains why the values tend to be higher than typical RE24 tables.

With all my run expectancy values in place, I executed the logic necessary (Appendix F.3) to obtain the RE Safe, RE Out, and RE Stay values. I then used the math mentioned in the introduction to compute the safety probability threshold for each play.

Results and Analysis

Now I could easily compare whether or not the play's safety probability exceeded the threshold probability and determine if the runner should have attempted an advance. Here are some examples:

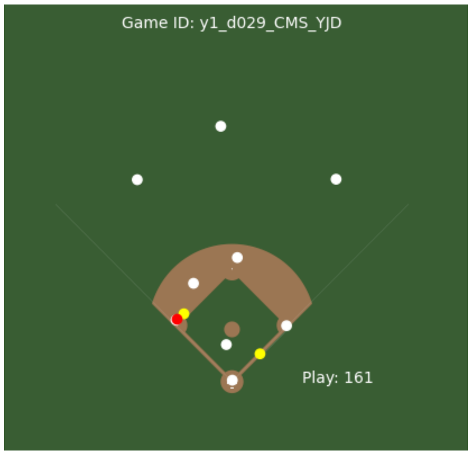


Figure 3: Click for GIF - Play has 3.7% model chance and a 9.0% breakeven chance for the 3B runner to be safe. This means he should not have gone; he did and was out.

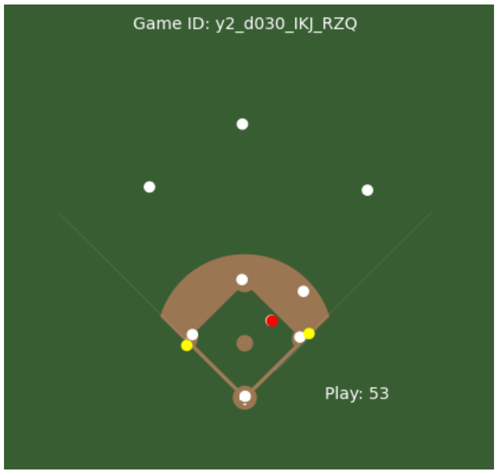


Figure 4: Click for GIF - Play has 99.9% model chance and 10.9% breakeven chance for 2B runner to be safe. This means he should have gone; he did and was safe.

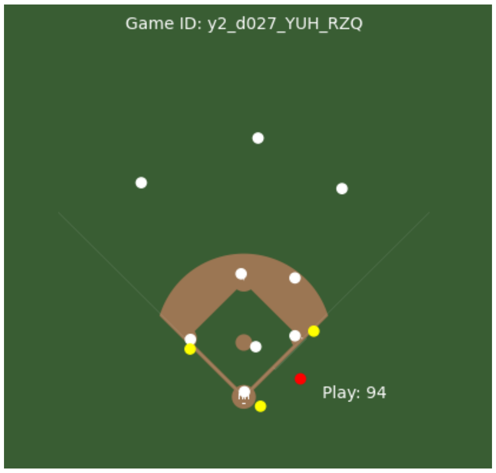


Figure 5: Click for GIF - Play has 29% model chance and 47.7% breakeven chance for 2B runner to be safe. This means he should not have gone; he did and was safe. Looking at the play, do you think he got lucky or had high BRI?

Comparing the theoretical advice on if they should have gone to their actual decision allowed me to classify the runner's choice as either smart (player did what BRI model said), aggressive (shouldn't have advanced but did), or passive (should have advanced but stayed).

From a general standpoint, we can see if there are any trends with where these baserunning choices are made. Figure 6 shows the choices and where they occurred based on where the fielder stopped the ball.

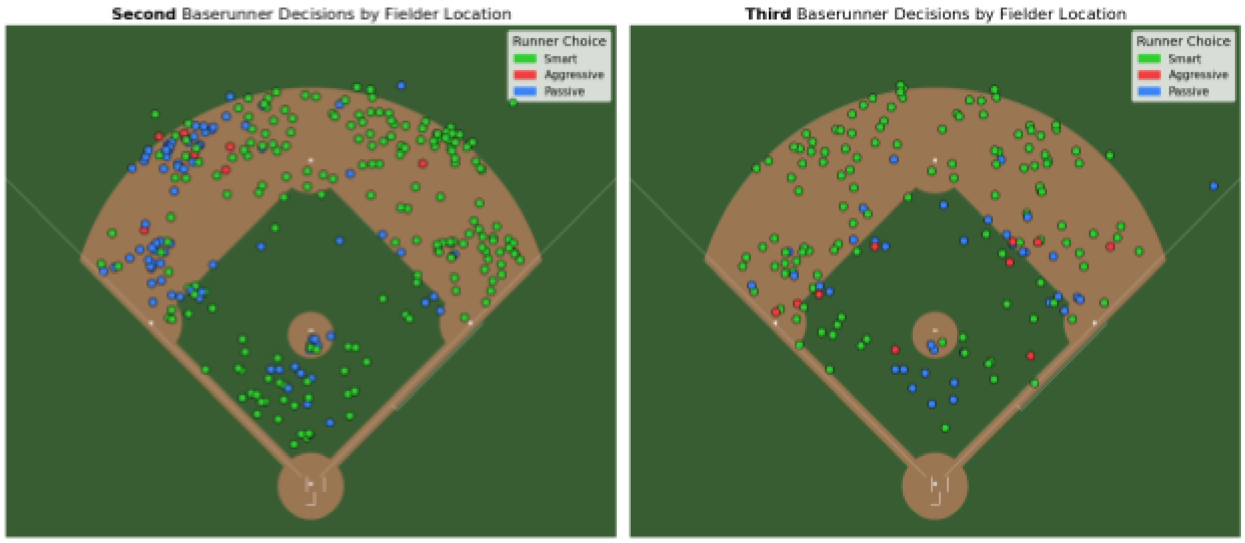


Figure 6: Comparison between runner choices for second and third baserunners. Dots indicate where the fielder stopped the ball. We can see that for 2B runners, both aggressiveness and passiveness are more frequent deeper in the infield and good choices are more frequent in front of the pitching mound and in between 1B and 2B. Contrarily for 3B runners, both passiveness and aggressiveness happen more often in shallow infield.

The smart choices are generally away from the base the runner is on and the aggressive plays are generally closer to the base the runner is on which makes perfect sense, giving us confidence that we can trust the other classifications as well.

Zooming in to a more personal analysis, we can also observe players' and teams' tendencies. The data tracked five teams for two years, so Figures 7 and 8 will focus on those teams.

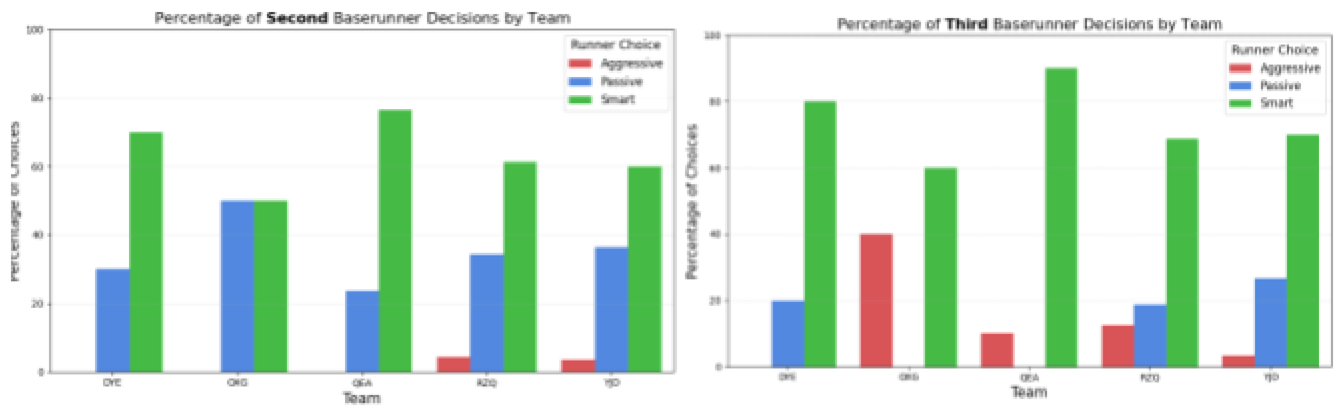


Figure 7: Differences in choice proportions among teams for second and third baserunners.

It can be seen that teams are more aggressive with runners on third base, sending them home on plays where they likely don't have the best odds to make it safely. Psychologically, I can see why because it could be very tempting to want to charge home for the score, so it's mentally easier to take that risk than to take a risk to simply advance and not score. Also if a ball is hit in the deeper part of the infield, the throw is farther home than to 3B, so perhaps runners think they have a better chance.

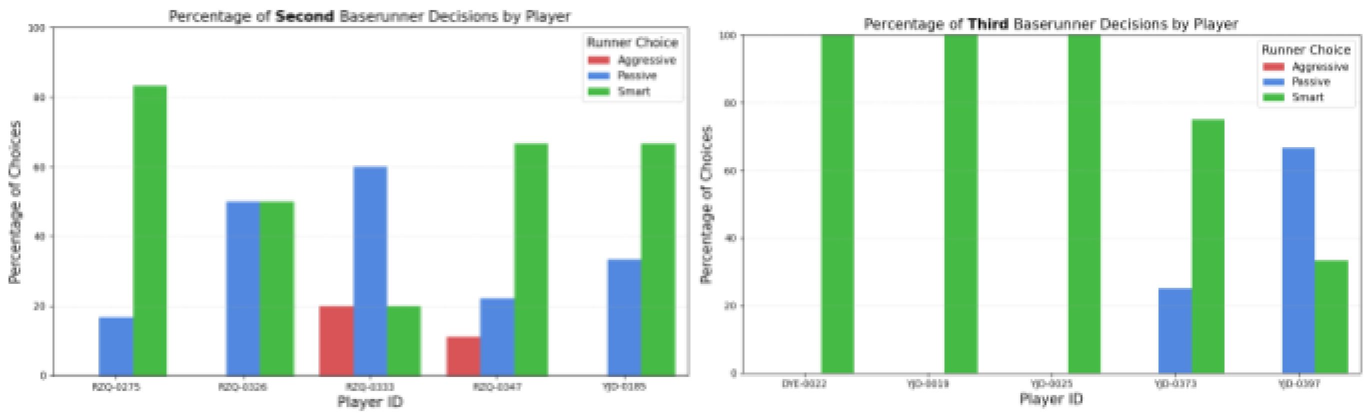


Figure 8: Differences in choice proportions for the players with the most advancement opportunities, separated by second and third baserunners.

Because many games had to get thrown out due to a lack of quality, there appeared to be a couple teams whose data remained clean throughout, leading to only a couple of teams being represented here. The charts are more interesting for the second baserunners as we see some more variance, but the third baserunners have three identical distributions and the remaining two are almost exact opposites which is interesting. Just like with the team distribution, I would be even more curious to see how this would look on a large scale dataset with more plays of interest and less plays with missing values. These visualizations are key to understanding baserunning intelligence because they are simple visuals that tell a much larger story: in quick-thinking moments certain players have a better understanding of when they should or shouldn’t go and having these types of players on a team that plays smart, fundamental baseball is only increasing their chances of winning. And to add to that, playing with good baseball IQ is not an easy metric to track so it becomes one of the hidden valuables to look for that can improve team success while saving money because the hidden metrics are much cheaper.

Conclusion

The purpose of this entire project was to give meaning to the probability thresholds that already exist by creating a model that estimated the probability of safety on a given play. On the other hand, if models do already exist that estimate the probability of safety, it's useless without a threshold probability to compare it to. You can't have one without the other because without both, there is no context for any meaningful evaluation. The only way to truly respect the game is to approach its problems with nuanced solutions and explain them in appropriate contexts. So the next time I'm fielding ground balls with my dad, I'll just crunch the numbers to see if it's even worth trying holding the runner.

Acknowledgments

Thank you so much to SportsMedia Technology for hosting this competition, sharing their valuable resources and time for us students to have such a fulfilling experience. Big thank you to Dr. Meredith Wills for communicating everything so well and being so available throughout the entire challenge. Another massive thank you to Billy Fryer for helping us out answering questions and keeping us entertained with your memes! Shoutout to Matthew Tokunaga and Triton Ball Club for sharing the link for this challenge. Thank you Eddie Dew for creating and sharing your animation code, it made the process so much easier and more fun. And last but not least thank you to my dad for helping inspire this project, letting me explain it to you, and helping me come up with the project name.

References

Reference #1 - (2025, June 16). Bless You Boys.

Reference #2 - Kagan, D. Society For American Baseball Research

Reference #3 - milb.com

Reference #4 - MLB Film Room

Appendix

Appendix A: Future Work and Limitations

I know there are things I may have missed when putting innings and outs together, and I would have loved to have experimented with other models to ensure XGBoost was the right decision, but like I've already said, the proof of concept is there and its strong, so I am content.

I would like to think that I've laid some foundation for something that could be even more impressive given resources that are more apt for this idea. Data that officially includes outs and the safety results of a play would make any findings extremely more accurate and likely more interesting. It would especially be interesting if there was more accurate data for individuals so players could be evaluated with much larger sample sizes and so we could see if there are larger choice distributions.

Another area of improvement is definitely the run expectancy values. I think the logic is very sound for calculating the RE24 matrix, however there is no way my code is the most efficient as it took about 15 minutes to gather data for only 2 months of baseball when it only took 2 minutes to get 1 month of data. I don't have any other times to compare this to but the time complexity here is obviously growing exponentially. If I could get data from 1-3 years, that would improve the accuracy of the run expectancy values and therefore make the decision evaluations more accurate.

Appendix B: Cleaning and Imputing Data

Technical Detail #1: Inferring Innings

I knew if I wanted to get outs, I would need to start with fixing the innings which was very difficult in the beginning because the only inning indicator was if it was the top or bottom of the inning, so there would be several rows were it seemed like an entire half inning or game was removed, so what I decided to do is tag the rows that did seem accurate, meaning they had the same top/bottom value for both the previous and next row in the same game. If they had the same values, I would give it a “same” tag, but if it was the same game and the previous row was different than the current row, I gave it a “switch” tag and the rest were tagged as null values. This was done so I could see if the difference in timestamps was significant enough to be predictable. Fortunately it was.

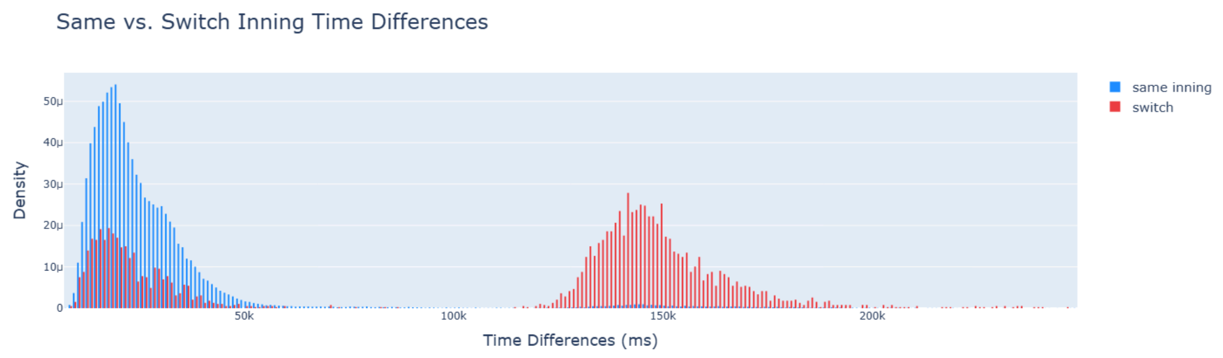


Figure 9: Density histogram showing the time differences for same vs. switch innings. There clearly exists a bimodal distribution for the switch values, which could make it hard to predict.

My original idea was to classify the missing inning values as either “switch” or “same” based on a hard threshold because the experiment for this was quite accurate.

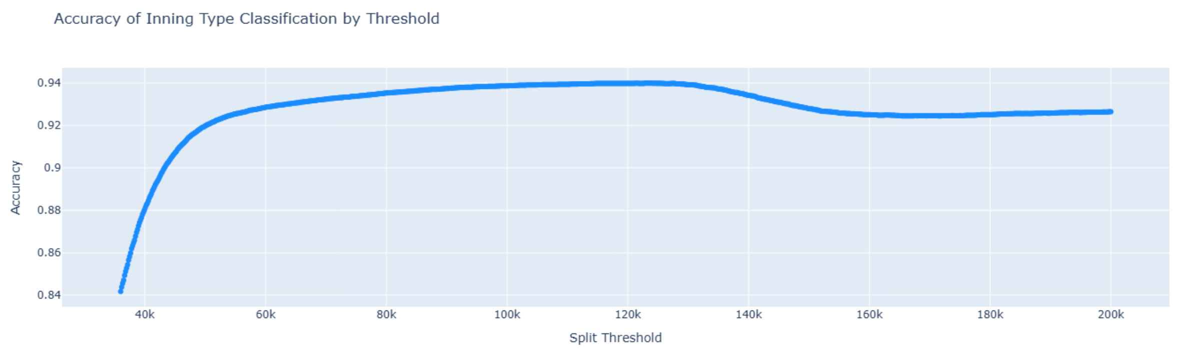


Figure 10: Scatter plot showing the accuracy scores based on a hard threshold. Used on only the data that had no missing inning type (switch or same) values. The highest accuracy seen was 94% and that was by classifying a play as having switch innings if its time difference from the previous play was 123007 ms or greater.

This was a hopeful idea but I quickly started seeing too many examples that looked wrong, so I moved to a random forest model, so it could hopefully pick up on the non-linear patterns. I ended up getting a 96% weighted average f1-score which was the most important metric to use for evaluating the model because the distribution between "switch" and "same" plays was very imbalanced.
I then used the model to predict the play's inning type which led to me assigning inning IDs to every play by increasing the ID whenever "switch" was seen. This would allow me to get rid of innings that were missing too many plays because those innings would be extremely hard or even impossible to count the number of outs for.

Technical Detail #2: Updating Baserunner Occupancies

While exploring the data by using the animation code, I quickly found out that many of the plays' info data said there was a man on base or not a man on base when the positional code contradicted those statements. Because the positional data actually showed the runner moving or not moving, I decided to replace all rows where the game information data contradicted the positional data with a boolean indicating if the runner was on base or not, according to the positional data. This explains why there are so many missing plays for individual players.

Technical Detail #3: Filtering Innings of Interest

Now that I have reliable base state data and somewhat reliable inning data, I was able to reduce the size of my dataset by filtering to only the innings that had a play of interest in them. This got rid of a lot of rows that were missing which was a relief.

Technical Detail #4: Inferring At-Bats

Because outs are impossible to count without knowing if there was a change in at bats, I needed to fill in these missing at bats. Using an almost identical method to inferring innings, I decided to predict the difference in the at bat number to see if there was a new batter at the plate. I once again used a random forest model inputting the time difference from one play to the next and landed on a 79% weighted average f1-score which was also the most important metric to look at because the distribution was imbalanced. Once I predicted the difference in at bats, I assigned an at bat ID to each play using the same method as the inning ID strategy.

Technical Detail #5: Calculating Outs

To count outs, the only way I could do it was manually. Again, I assigned each play and outs difference by going play by play and checking all the possible scenarios I could think of that would cause an increase in outs, or a confident scenario where no outs would be gained. I also gave each play an outs confidence number to help improve the final model by giving it a clue that there are times when outs are not a reliable feature. After calculating the difference in outs for each play, I finally pieced together the number of outs for every play and was able to apply the last filter to my plays of interest.

Appendix C: Feature Engineering

Technical Detail #1: Calculating a Rate of Change

Whether it was arm strength, exit velocity, or sprint speed, I took either the positional or ball tracking data and took the euclidean distance between the current and previous timestamp for all timestamps I had available within that play. This is probably the most standard and simplest way, but it got the job done and proved to be useful in the model.

Technical Detail #2: Determining if a Runner Was Safe

Once I had all my plays filtered, I then looked at the next play to see if the runners had advanced and if they were safe. If there was no next play available, I got rid of the play of interest because it likely means there were two outs, so it should have never been a play of interest in the beginning. I realize there could have been a double play as a result of one of the runners getting thrown out, but that would have been too complicated to figure out for the scope of this project.

For plays that did have the next play available, I could determine if the second baserunner was safe, but I had to use some geographical and timing inferences to determine if the 3B runner got home before a throw to the catcher did. If there was no throw to the catcher after the ball was put in play, I automatically tagged the 3B runner as safe. Because of these reasons and the lack of outs accuracy, 90% of runners were safe.

Appendix D: Number of Plays of Interest Justification

The entire dataset originally consisted of a little less than 80,000 plays so it was surprising to see that only about 380 plays made it through the filtering. That was until I did some side research. I randomly selected 5 MLB teams (because SMT data was collected for 5 random MiLB teams) and went on MLB Film Room (Reference #4) to query the play scenarios I'm looking at from 2024-2025 (accepting we're midway through 2025 because there are less MiLB games than MLB). While the query system also adds in plays that I don't want, it's close enough to make the comparison. Between the 5 teams, there were roughly 470 results which include bases loaded scenarios and have more games to choose from. This validates the size of my dataset and provides evidence that I calculated innings and outs somewhat accurately.

Appendix E: Model Selection

I chose XGBoost for my safety probability model because it is one of the most well-regarded statistical models that can pick up on non-linear patterns. Also its ability to detect missing values as a legitimate feature was a key component to its strength. Rather than ignore them or force the engineer to remove them, (and lose multiple rows of good data) the model can learn from the missing values. This was important because on some plays, the fielder never attempted a throw which is a major indicator that the ball was hit in a way that made throwing out the runner very hard or even impossible, so having arm strength as a missing value played a huge part in increasing the accuracy of the model.

Appendix F: Run Expectancy

Technical Detail #1: Base-Out States

Run expectancy requires the knowledge of the base-out states. Base states can be represented using 3 digits where the first digit represents 1B, the second 2B, and the third 3B. The value of each digit indicates if the base is occupied or not. For example a man only on 2B is represented as 010, or you can also see it as 020 to emphasize the 2 is 2B. There are 8 base states (000, 100, 010, 001, 110, 101, 011, 111) and 3 out states (0, 1, 2), making 24 total combinations of base-out states. An example of a base-out state would be man on 2B with 1 out which can be represented as 010, 1 out.

Technical Detail #2: How to Interpret Run Expectancies

At any point in a baseball game, the game will be in one of the 24 base-out states. What a RE24 matrix tells you is how many *additional* runs a team can expect to score from now to the end of the inning. For example, looking at Figure 2, if a team is winning 1-0 and they have a man on 2B with 1 out, a team can expect to score 0.66 more runs by the end of the inning bringing their total lead to 1.66-0. Obviously this isn't possible but it's an accepted way to calculate risk and reward.

Technical Detail #3: How to Calculate Safety Probability Threshold

For this project's analysis which looks at infield ground balls where the 2B or 3B runner has a choice to advance or stay, we are only looking at 3 possibilities for each runner, independently. A runner can either:

- Advance and get thrown out
- Advance and be safe
- Stay put

To use the formula included in the introduction, we need the REs of being safe, out, and staying put. This is a major assumption I am going to make and it could be critiqued, but for the sake of this project, it will do just fine. I am going to assume that if the runner tries to advance, the fielder will attempt to throw him out as part of a fielder's choice, meaning the batter is safe at 1B. Let's take the example of the base-out state of 011, 1 out. I am also looking at each runner independently, so here let's look at the third baserunner. If the runner advances and is thrown out, then the next base-out state is going to be 101, 2 outs because both the batter and the second baserunner were safe on advancement. Looking at Figure 2, this base-out state gives us an RE of .583. If the 3B runner is safe, the next base-out state is 101, 1 out because we are assuming the fielder threw home but not in time. This gives us an RE of 1.085. Finally our base-out state of staying put is just 011, 2 outs which is a 0.64 RE. If we want to see what the probability we need to be safe to make it worth advancing vs. staying put, we use the formula,

$$\text{RE Safe}(p)+\text{RE Out}(1-p) = \text{RE Stay}$$

plugging in using the values we got, we end up with:

$$1.085p + 0.583(1-p) = 0.64$$

and after using algebra to solve for *p*, we get *p* = 0.11 which means we need at least an 11% chance of being safe in order to increase the number of runs we can expect to score.