

Predicting Barrels From Pitch Characteristics to Examine Pitcher Mistakes

Justin Ford

2/8/2022

What makes a mistake?

You know it when you see it: the batter's eyes light up, the pitcher's gaze drops to the ground just as the bat connects. A meatball. Even casual baseball fans probably have some familiarity with the concept of a so-called "mistake pitch." The breaking ball that doesn't *quite* break enough, the fastball right down Broadway. Ultimately, I think the simplest definition of a mistake pitch is a pitch that's not just likely to get hit, but likely to get *crushed*. Of course, not every pitch that gets crushed should be classified as a "mistake," nor is every inaccuracy properly punished by the hitter. Sometimes, a batter puts a good swing on a good pitch. Then how can we discern if a pitch is likely to get crushed or not? What makes a pitch a mistake pitch?

The first step is to find out what exactly "crushed" means. There are a few ways to do this:

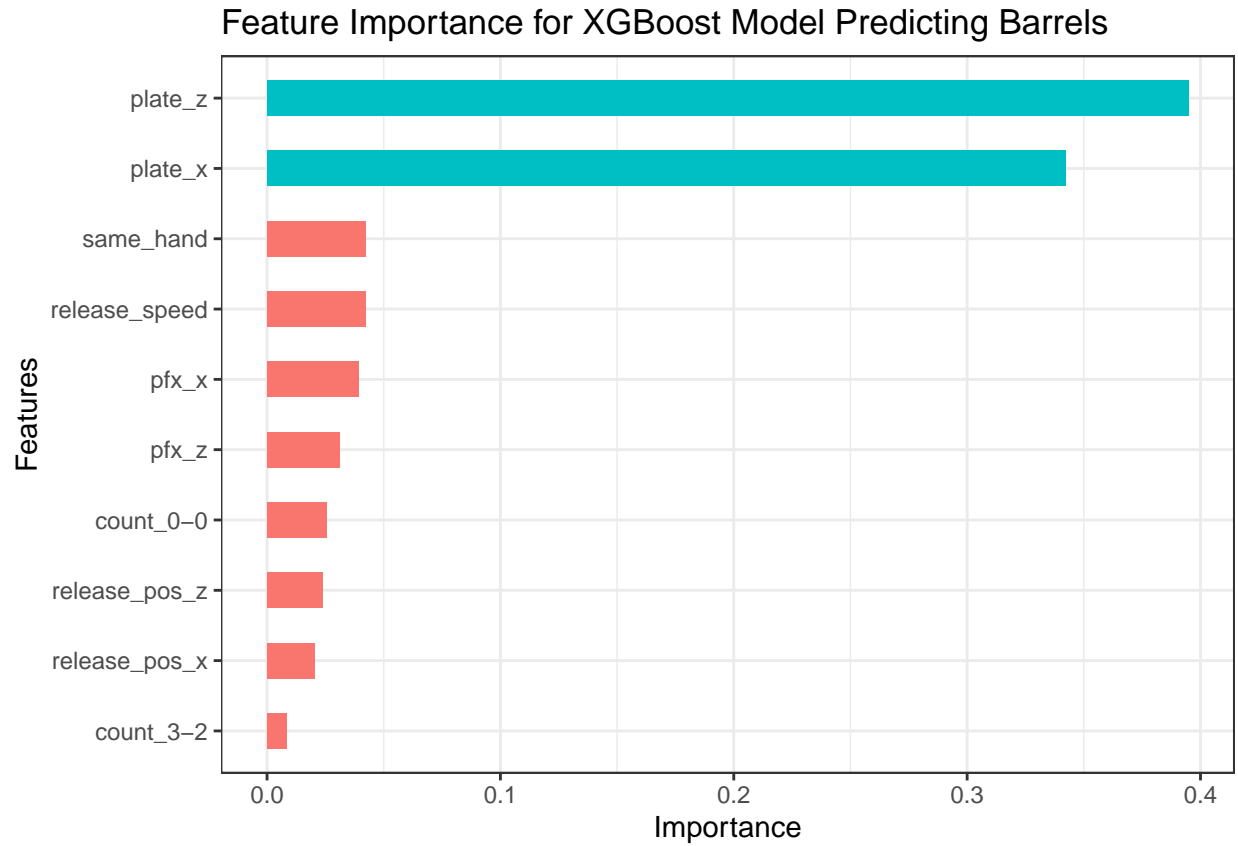
- Set a threshold for run value, so that all batted balls over the threshold are considered crushed. However, run values don't always correlate closely with how "well-hit" a batted ball is. Bloopers tend to have a high run value because they fall for hits often, but wouldn't be considered to be hit squarely or hard.
- Use Statcast's Barrel classification as a proxy for "crushed." This is the simplest and most direct option, because these are literally the batted balls that Statcast defines as "hit well." The one issue with using barrels is that batters will have widely differing abilities to hit barrels. Some hitters, like David Fletcher or Myles Straw, will rarely ever have a batted ball classified as a "Barrel," even if they hit it as well as they can. Does that mean it's impossible to throw a mistake pitch to David Fletcher? I don't think so.
- Use a subset of the "best" batted balls by each hitter. This solves the problem with using Statcast Barrels. However, I would need to devise a system to come up with evaluating with batted balls are the "best," or rather, "most crushed." The best way I could think to evaluate batted balls is by using run values or the built-in "estimated_woba_using_speedangle" column of the Statcast csv. Of course, this then presents us with the same problem we saw in option 1; estimated_woba_using_speedangle isn't necessarily a measure of how well a ball is hit in the same way that Barrels are.

For simplicity's sake, I decided to just use Barrels as my stand-in for "crushed." It is the most direct measure of what I mean by "crushed" that I have readily available.

So, looking at every pitch from the 2021 season (data retrieved using the excellent `baseballr` package for scraping from Baseball Savant), I constructed a model that would predict the probability that a given pitch would be barreled.

The features of my model included pitch velocity, location, movement, pitcher release point and the count. My model does not include any information regarding batter tendencies, or any contextual information about the pitcher's arsenal. It is simply looking on a pitch-by-pitch basis, regardless of batter, and trying to determine how often that pitch gets barreled on average.

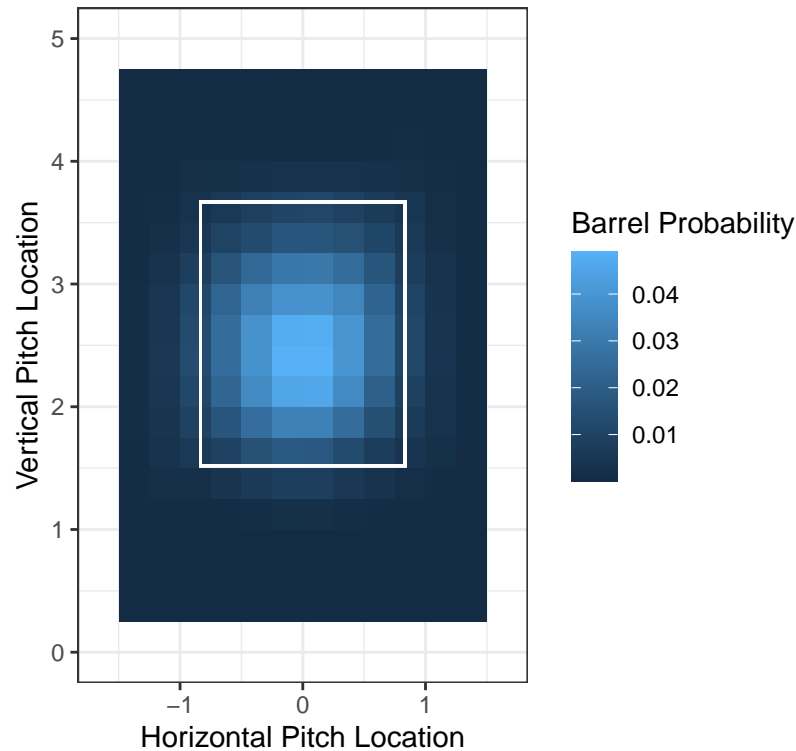
After training the model, we can look at what features of the data contribute the most to correctly predicting whether the pitch was barreled or not.



A more complete description of what some of these attributes are measuring can be found [here](#), but essentially `plate_z` and `plate_x` refer to vertical and horizontal location respectively. These are by far the most important features for them model. This is probably as expected, though I would've expected movement (`pfx_x` and `pfx_z` are the measurements for movement) to be a bigger contributor.

We can bin pitches by their location and look at the average barrel probability for pitches in each bin.

Barrel Probability by Pitch Location
Strikezone From Catcher's Perspective



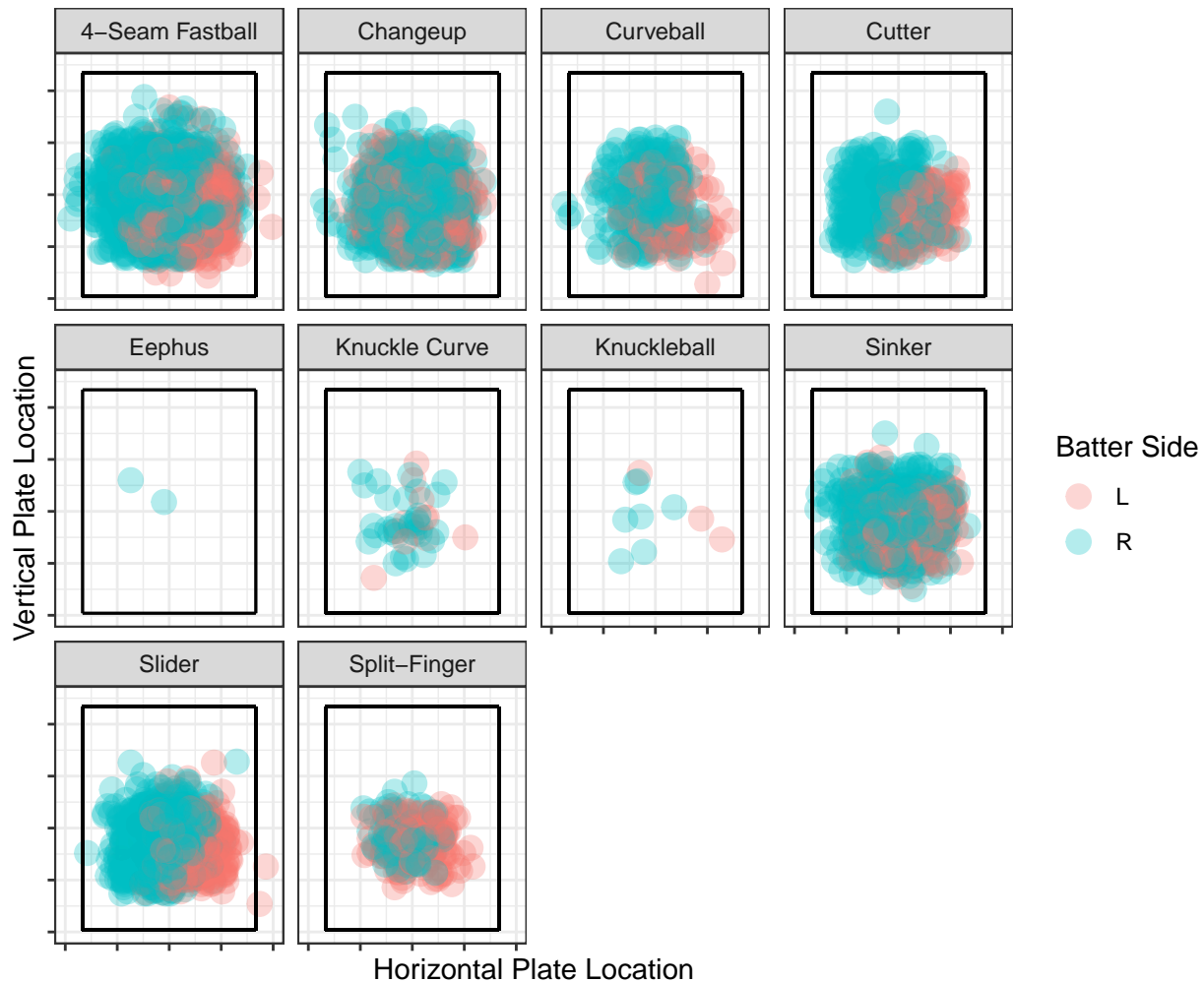
As we would expect, pitches nearest the middle of the zone have the highest probability of getting barreled according to the model.

Pitcher Mistakes

Since we have the probability that every pitch will be barreled, we can plot the pitches with the highest barrel probability (these are our “mistakes”). I’m taking the top 1.5% of pitches to be our “mistakes,” since this is roughly the percentage of pitches that are barreled league-wide.

Mistake Pitch Locations By Pitch Type, 2021 MLB Season

Catcher's Perspective



Once again, we see how many mistakes are pitches that are left over the middle of the plate, for all pitch types. We can also see how more mistakes tend to be made on the inside part of the plate. Also, breaking balls and offspeed pitches up in the zone don't really show up here as "mistakes," as maybe conventional wisdom would suggest.

Which Pitchers Avoid Mistakes?

Now that we have a way to quantify how much of a mistake a pitch is (barrel probability, which I will also call xBarrels), we can look at the pitchers who are the best and worst at avoiding these pitches. However, I don't just want to look at who throws the most mistake pitches, I want to see which pitchers are able to prevent hard hits while also throwing pitches that get lots of swings and called strikes. A pitcher who

throws 55 foot curveballs every pitch technically isn't throwing any mistake pitches by our definition, because those pitches are essentially impossible to barrel. Those pitches, though, have little chance of accomplishing anything positive for the pitcher's team. So, I trained another model which attempted to predict the likelihood that a pitch would either be: a) swung at, or b) called for a strike. I call this likelihood "xChance" (i.e. the "chance" that the pitch results in a strike or batted ball). This way, I'm rewarding pitchers that are able to throw strikes *and* avoid mistakes.

So we have a measure for the number of "mistakes" a pitcher made (xBarrels), and a measure for the number of "chances" that a pitcher allows for batters to barrel a pitch. We can simply take the proportion of xChances that end up as xBarrels for each pitcher, and we will have a good measure for determining pitchers that throw highly barrelable pitches:

Top 10 Barrel Prone Pitchers, 2021 MLB Season

1500 pitch min.

Pitcher	Pitches	Chances	Barrels	xBarrels (xB)	xChances (xC)	xB/xC
Ross Stripling	1678	1065	33	33.44	1,090.85	0.0307
Kris Bubic	2198	1364	36	42.35	1,393.34	0.0304
Caleb Smith	2055	1283	31	39.00	1,291.62	0.0302
Marco Gonzales	2319	1497	60	44.64	1,500.81	0.0297
Eric Lauer	1938	1238	25	35.79	1,247.46	0.0287
Kolby Allard	2048	1338	46	38.65	1,348.30	0.0287
Keegan Akin	1762	1152	35	33.22	1,161.28	0.0286
Austin Gomber	1840	1200	34	34.01	1,195.96	0.0284
Tyler Anderson	2573	1775	48	50.07	1,770.50	0.0283
Zach Plesac	2200	1451	48	40.58	1,435.21	0.0283

And the pitchers who are the best at throwing hard-to-barrel pitches:

Top 10 *Least* Barrel Prone Pitchers, 2021 MLB Season

1500 pitch min.

Pitcher	Pitches	Chances	Barrels	xBarrels (xB)	xChances (xC)	xB/xC
Luis Castillo	3139	1993	25	30.75	1978.7602	0.0155
Logan Webb	2205	1432	21	24.33	1428.5178	0.0170
Lance McCullers	2784	1707	27	30.19	1753.2585	0.0172
Gerrit Cole	2966	1972	46	33.44	1925.5431	0.0174
Corbin Burnes	2574	1702	18	29.96	1619.6559	0.0185
Charlie Morton	2967	1911	25	36.00	1940.5631	0.0186
Sandy Alcantara	3077	2083	39	38.84	2070.2882	0.0188
Aaron Nola	2957	1993	38	37.48	1972.5812	0.0190
Alex Cobb	1574	997	11	19.39	985.5028	0.0197
Zack Wheeler	3179	2138	30	42.42	2152.0332	0.0197

We can also look at what individual pitches are thrown for "mistakes" the most often.

Top 10 Mistake Prone Pitches, 2021 MLB Season
500 pitch min.

Pitcher	Pitch	Pitches	Chances	Barrels	xBarrels (xB)	xChances (xC)	xB/xC
Cole Irvin	4-Seam Fastball	1086	759	19	26.68	775.1186	0.0344
Austin Gomber	4-Seam Fastball	744	489	20	16.77	492.7852	0.0340
Ross Stripling	4-Seam Fastball	854	558	19	19.39	570.3857	0.0340
Caleb Smith	4-Seam Fastball	991	607	20	20.73	618.5581	0.0335
Michael Pineda	4-Seam Fastball	917	687	21	22.42	675.0106	0.0332
Kris Bubic	4-Seam Fastball	1142	704	14	24.08	728.6825	0.0330
Lucas Giolito	Changeup	938	666	8	21.19	642.9885	0.0330
Rich Hill	4-Seam Fastball	1165	829	19	26.84	815.3575	0.0329
J. A. Happ	4-Seam Fastball	1437	945	31	29.86	924.3057	0.0323
Dean Kremer	4-Seam Fastball	547	360	13	12.11	378.6366	0.0320

An interesting one here is Lucas Giolito’s changeup, which was actually a very good pitch for him last season (-11 run value according to Baseball Savant), but is significantly outperforming my xB measure. It’s not clear to me whether Giolito simply got lucky with his changeup last year (it also graded out well in 2019 and 2020 my run value so this doesn’t appear to be very likely), or if there’s some attributes of the pitch that aren’t captured by my model. Most of the rest of the pitches in this group are fastballs from pitchers with below average velocity, which is the type of pitch I would expect to see get barrelled often.

And finally, the ten pitches least likely to be “mistakes” in 2021.

Top 10 *Least* Barrel Prone Pitches, 2021 MLB Season
500 pitch min.

Pitcher	Pitch	Pitches	Chances	Barrels	xBarrels (xB)	xChances (xC)	xB/xC
Tyler Rogers	4-Seam Fastball	599	456	1	3.87	451.5998	0.0086
Emmanuel Clase	Cutter	731	503	3	6.18	495.6977	0.0125
Luis Castillo	Sinker	748	507	9	7.03	517.3606	0.0136
Miguel Castro	Sinker	523	297	7	4.15	305.2986	0.0136
Sonny Gray	Curveball	510	300	2	4.16	304.1963	0.0137
Sandy Alcantara	Sinker	865	615	6	8.48	616.9510	0.0137
Logan Webb	Slider	611	386	5	5.17	365.9697	0.0141
Luis Castillo	Changeup	959	624	4	8.67	610.4705	0.0142
Charlie Morton	Curveball	1091	708	4	10.21	712.6517	0.0143
Jeury Familia	Sinker	594	413	8	5.90	411.1640	0.0143

Conclusion

Now we have a bit of a better sense of what makes a mistake pitch, even if what we found out mostly aligns with conventional wisdom. Pitches in the heart of the plate are the most likely to get damage inflicted upon them, and hitters tend to barrel more pitches that creep onto the inside half of the plate. Our model also showed that breaking balls and offspeed pitches up in the zone are not as punished as one might think.