Project 3: Predicting Seam Shifted Wake

A Kaggle Inspired Project

By
Eddie Dew

November 1st, 2024

Introduction

When a projectile is thrown, outside forces are acting upon it causing the object to move. For the instance of a baseball, there were thought to be two components: gravity and the magnus effect. Gravity affects the movement of the baseball vertically, making it move on a downward trajectory for any pitch. The magnus effect is the displacement of air created by a spinning object, which is the case for many pitches thrown in baseball. Think about a duck swimming on a pond with a ripple of water following behind it. That's essentially what the magnus effect is in a fluid space, moving through the air and causing it to intensify the trajectory that is already caused by gravity. For example, a fastball is spinning with a bunch of backspin, causing more air displacement towards the bottom of the ball, causing it to not break as much as expected if gravity was only at play, creating a "rise" effect, which is not truly the case but it appears that way. The opposite is true for a curveball. These were the only factors to explain movement until a study conducted in 2020 at Utah State University by Dr. Barton Smith and his team of Engineers. They studied how seam orientation relative to the axis of the ball alters the movement of the pitch. A way to think about it is the two previous modes of movement are only true with a ball that's perfectly smooth; a baseball has seams on it which makes it not a perfectly smooth object. Their study concluded that the orientation of the seams and relative spin axis can have a significant impact on the movement of the pitch if thrown in a "gyro" plane called Seam Shifted Wake (SSW) (Smith, Andrew W. 2020 p. 26). In this paper, SSW will be predicted using regression and analyzed to find the strengths and weaknesses of the metric.

Dataset Description

The dataset comes from Baseball Savant, a website that has tracking data from all over the league. What this paper will be specifically targeting is their pitch level data which has the

characteristics of the pitch thrown using high speed tracking cameras. The data is scraped from a python package called Pybaseball. Dates scraped ranged from the Major League Baseball 2020 to 2023 regular seasons.The dataset comprises 2,449,241 pitch observations with 92 columns. It has rows with NA values, typically ones where the pitch was so badly thrown, it couldn't be picked up by the tracking system. The target features, horizontal (pfx_x) and vertical break (pfx_z), have 17 and 6 NA columns respectively. Pitches are oriented for both right and left handed pitchers so to make it easier for the model to understand, all statistics on the horizontal plane were converted to a right handed trajectory. A feature of interest that wasn't in the dataset was spin efficiency. Luckily, a paper written by Alan M. Nathan called *Determining the 3D Spin Axis from Statcast Data* (2020) derives this feature for us using physics and the data from the Baseball Savant dataset.

Exploratory Data Analysis

The first feature used in this model is Spin Efficiency. As mentioned, it had to be derived from empirical research. Once derived, some calculations appeared off so only efficiencies between 0 and 1 were kept in the model training data. The next feature used was spin axis, or the angle the ball is oriented with the direction it is moving towards. The last feature used was release spin rate, how many rotations the ball has while in flight. All of these features had no NA values after the filter was applied. Since this paper is more focused on the model side, not much EDA was required with so little features. The idea behind this model is to predict horizontal and vertical break using the spin characteristics to capture SSW via expected movement. It should be noted that Spin Efficiency and spin axis have a bimodal distribution. Further analysis was not conducted however and will be for future work. Again, the main goal was to prioritize making different models on the target features.

Model

Three different Regression models were used to predict horizontal and vertical movement. The first was Linear Regression. Running a linear regression model using OLS least squares helps gain insight about what features are influential and what features are significant. Both models found spin efficiency to be the most impactful feature in terms of correlation (-0.7313 and 0.2797 respectively). For context, a negative correlation indicates a lower spin efficiency generates more horizontal movement, the exact conclusion denoted in Dr. Barton Smith's paper. A moderate positive correlation indicates that higher spin efficiencies generate more vertical movement. Both make sense. When looking at the p-values, all features were statistically significant with the critical value being 0.05. After running this basic test, the 3 different models were tested. Figure 1 and 2 denote a table of the results for each model of each target feature where a baseline model is taking the average of the respective target feature.

Metric	Baseline	Linear Regression	XGBoost	XGBoost KFold (10 Splits)
R^2	0	0.739	0.918	0.919
MSE	0.631	0.165	0.052	0.051
RMSE	0.795	0.406	0.227	0.226
MAE	0.682	0.314	0.167	0.166

Figure 1: Model Results for Horizontal Break

Metric	Baseline	Linear Regression	XGBoost	XGBoost KFold (10 Splits)
R^2	0	0.508	0.871	0.871
MSE	0.525	0.259	0.068	0.068
RMSE	0.725	0.508	0.261	0.260
MAE	0.593	0.391	0.192	0.192

Figure 2: Model Results for Vertical Break

Just from jumping from a baseline model to a Linear Regression model almost doubles the accuracy. XGBoost does an even better job at capturing the nuances of the break data. This makes sense as XGBoost uses gradient boosted decision trees to make decisions on the predicted movement value, much more effective for a tabular dataset. It's also able to bypass the bimodal distributions for the spin efficiency and spin axis features, something Linear Regression does not perform well with. Running XGBoost through folds and increasing the number of estimators had very little impact on the error metrics.

Impact

Different models were used to break horizontal and vertical break. XGBoost appeared to be the best model out of the rest in its robust nature. Using this model, expected horizontal and vertical movement are computed. Taking the difference from the observed and expected breaks gives you SSW. One interesting takeaway that was found was applying SSW in context to run prevention. When you take the delta sum (abs(SSW_vert) + abs(SSW_horiz)) of the vertical and horizontal SSW values, you can analyze how much total break the pitch changed with respect to SSW. When you fit a polynomial regression plot of this delta sum in relation to the run expected value, you get Figure 3.

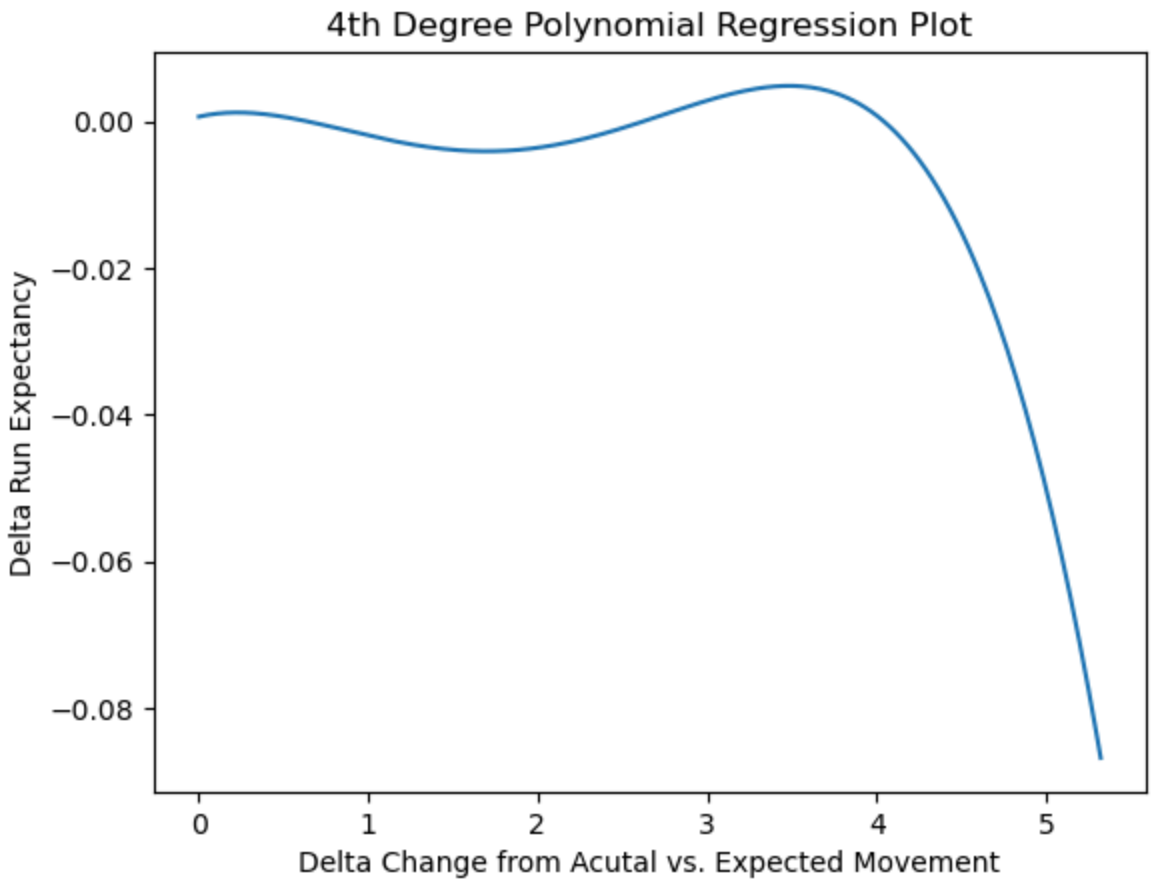

Figure 3: SSW in context of Run Prevention

To interpret this plot, the x values resemble the delta change of SSW, meaning that the higher the value, the more SSW that pitch has. On the y axis, delta run expectancy resembles the change in expected runs scored for that particular pitch where a negative number is optimal. From this plot, the sweet spot for an effective SSW pitch is to have a delta change between 1 and 3. There is also a spike past a delta sum of 4 however. It must be noted that there are not many values past 5 which means this regression equation is generalizing that last part of the line for a few data points, so it shouldn't be interpreted as having a really high SSW is necessarily a good thing.

In conclusion, this project attempted to quantify SSW from empirical studies using different Regression models. While more tests need to be done, this project can be used by MLB

teams to identify what pitchers have effective SSW pitches and try to leverage that for their success. This metric paves the way for pitchers who may not be genetically gifted with high velocity and spin rate to find success through SSW. Some things this paper could improve on are finding reasons why spin efficiency isn't always from a 0 to 1 scale such as incorrect calculations or bad data points. Another area this paper didn't explore is tuning the parameters of the model more nor explore techniques that can improve the Linear Regression model such as transforming the features. Maybe adding in pitch velocity could help improve the results. Overall, this paper has some foundation to implementing a metric that can identify pitches that are deceptive to the batter's eye.

References

LeDoux, James, and Moshe Schorr. “Pybaseball: Pull Current and Historical Baseball Statistics Using Python (Statcast, Baseball Reference, Fangraphs).” *GitHub*, github.com/jldbc/pybaseball

Nathan, Alan M. “Determining the 3D Spin Axis from Statcast Data” *University of Illinois*, 2020, baseball.physics.illinois.edu/trackman/SpinAxis.pdf

Smith, Andrew W., "Pitched Baseballs and the Seam Shifted Wake" (2020). *All Graduate Theses and Dissertations, Spring 1920 to Summer 2023*. 7903. https://digitalcommons.usu.edu/etd/7903