Building Machine Learning Models to Predict the Outcome of a Pitch in an MLB Game - Iteration 1

Zachary Becker, Zachary Armand, Zhaofeng Li

Northeastern University

DS 5500 Capstone: Applications in Data Science

March 30, 2025

Understanding the Dataset

Source:

Where did the data come from (source)? What was the purpose of collecting this data? Is the data publicly available, or are there restrictions on its use? please provide links.

1.1 Origin of the Data

The primary source of the dataset is Baseball Savant, a platform that provides MLB Statcast data. Statcast captures detailed measurements of every pitch thrown in Major League Baseball (MLB) games, including pitch velocity, spin rate, release point, and batted-ball metrics (exit velocity, launch angle, etc.). Statcast data collection began being phased in around 2015, making it a relatively new but rich source of baseball analytics.

Supplementary Sources:

- Baseball Reference: Offers historical and season-level statistics for MLB players and games.

- FanGraphs: Known for advanced sabermetric data, player projections, and custom leaderboards.

- Retrosheet: Provides historical game logs and play-by-play data for MLB.

- pybaseball (GitHub): A Python package that programmatically scrapes data from Baseball Savant, Baseball Reference, FanGraphs, and other sources.

Baseball Savant Statcast Search
Baseball Reference
FanGraphs
Retrosheet
pybaseball (GitHub)

1.2 Purpose of Collection

Statcast data is collected by MLB to enhance and quantify in-game analytics. High-speed cameras and radar systems track the motion and characteristics of each pitch and batted ball event, enabling teams, analysts, and fans to gain deeper insights into player performance and strategy.

1.3 Public Availability and Restrictions

- Publicly Available: Baseball Savant's Statcast data is accessible via its website, where users can query and download CSV files.

- Restrictions: There may be rate limits or usage constraints, especially for large-scale or automated scraping. Commercial or extensive research usage might require special permissions or adherence to MLB's terms of service.

Structure and Metadata:

What are the key features (columns) in your dataset, and what do they represent? What is the size of your dataset? Provide statistical summaries and key metrics about your dataset. Is there an API available for your dataset?

2.1 Key Features

Statcast data at the pitch-level typically includes:

pitch_type: Categorical label indicating the pitch (e.g., FF for four-seam fastball, SL for slider).
release_speed: Pitch velocity at release (mph).
release_pos_x / release_pos_z: The horizontal and vertical release coordinates (in feet).
pfx_x / pfx_z: Horizontal and vertical movement of the pitch relative to a spinless pitch (in inches).
spin_rate: Pitch spin rate (revolutions per minute, rpm).
spin_axis: Orientation of the spin in degrees.
plate_x / plate_z: Location of the pitch as it crosses home plate (in feet).
launch_speed: Exit velocity of the batted ball (mph), if contact was made.
launch_angle: The batted-ball launch angle (in degrees).
events: Outcome of the at-bat.
batter / pitcher: Unique IDs for the batter and pitcher.

2.2 Dataset Size and Summaries

- Pitch-Level Rows: $\tilde{7}$00,000 to 750,000 pitches per MLB season, reaching millions of rows when combining multiple seasons.

- Columns: Typically 50+ columns per row.

For a single season, common numerical summary statistics might include:

- release_speed: 50 to 105, with an average around 88-92 mph.

- spin_rate: 500 to 3200+, with an average around $\tilde{2}$200 rpm.

- launch_speed: 30 to 120+, with an average around $\tilde{8}$8 mph.

2.3 API Availability

Baseball Savant does not officially document a public API, but pybaseball provides convenient interfaces to relevant endpoints. We plan to download the available data via the pybaseball package and then upload this data to a cloud-based database. We will then either query the online database directly via python and a SQL client or download the data to a .csv file.

Missing Data:

Are there any missing values? If so, how are they represented? What strategies can be used to handle missing data?

3.1 Representation of Missing Values

- Often coded as NaN in CSV files if radar or camera systems failed to capture a metric (e.g., spin rate, release point).

- Older data (pre-2015) might have incomplete coverage of advanced Statcast metrics.

3.2 Strategies for Handling Missing Data

Dropping Rows: If the fraction of missing data is small, removing those rows can be feasible.

Imputation: For numeric columns, replacing missing values with mean, median, or pitcher-specific averages.

Domain-Specific Treatment: In some cases, missing pitch metrics might be systematically older data, so limiting analysis to recent years can ensure more complete data quality.

Anomalies:

Are there outliers or anomalies? do they indicate errors? How consistent is the data across features and observations?

4.1 Detection of Outliers

Pitch Velocity: Values below 50 mph or above 105 mph may be errors or rare "eephus" pitches.

Spin Rate: Extremely high (e.g., 5000 rpm) or zero rpm can indicate measurement error.

Launch Angle: Angles beyond typical ranges (e.g., +90 or -90) could be data capture issues.

4.2 Consistency

For modern seasons (2019+), data is generally reliable. When merging multiple data sources, we must ensure consistent identifiers and date formats.

Bias:

Does the dataset adequately represent the population or system of interest? Could there be biases in the data collection process that might affect the analysis?

Certain pitchers or hitters may be heavily represented if they play more or if the dataset focuses on certain seasons or events. League-wide trends change year to year, which may skew results if multiple seasons are pooled without accounting for annual changes. Heavier weighting of high-volume players can skew model training outcomes. If historical data is combined with modern Statcast data, differences in measurement technology or league conditions can introduce systemic biases.

Distributions:

What are the distributions of numerical features? Are they skewed or normally distributed? Are there strong correlations between features? Could multicollinearity be a concern?

6.1 Numerical Feature Distributions

- Pitch Velocity: Slightly right-skewed, centered around 88–92 mph with a tail extending to high 90s and 100+.

- Spin Rate: Approximate mean $\tilde{2}200$ rpm, with a right tail up to 3000+ rpm.

- Launch Angle: Often near 10–15 degrees average, but can be widely spread with negative angles and high positive angle.

6.2 Correlations and Multicollinearity

- Velocity & Spin Rate: Modestly correlated; higher velocity often but not always ties to higher spin.

- Release/Movement Variables: Release extension, release point, and pfx_x/pfx_z can be interrelated.

- Concern for Multicollinearity: In linear models, highly correlated features can inflate variance in coefficient estimates. Tree-based models like XGBoost are more robust to multicollinearity, but interpretation of feature importance may still be affected.

Categorical Data:

How many unique categories exist for each categorical variable? Are the categories well-balanced or imbalanced?

Several categorical fields are unique ID or name fields. These include 'player_name', 'batter', 'pitcher', 'home_team', 'away_team'. These fields all have no missing values and are all populated for all rows of 2024 data. The count of unique values in these fields are seen in Table 1.

Column	Unique Count
player_name	1176
batter	1435
pitcher	1178
home_team	30
away_team	30

Table 1: Unique Classes for ID and Name Variables

Other categorical columns of interest and their respective unique value counts are listed in Table 2.

Some variables are imbalanced. Oftentimes, this correspond generally to the probably of these values occurring in a baseball game. For example, right-handed hitters are more common in the MLB (and general populace), and the Statcast field 'stand' reflects this fact. Histograms of the frequency of the different unique values for the categorical variables can be seen in Figure 1.

Ethical Considerations:

Does the data contain sensitive or personally identifiable information (PII)? Are there any ethical concerns in using or publishing insights from this data?

This dataset contains no personally identifiable information or other ethical concerns. The data is published by the MLB with the intent to be used by the general public.

Alignment with Goals:

Column	Count Unique
pitch_type	17
description	13
zone	13
hit_location	9
woba_value	7
n_priorpa_thisgame_player_at_bat	7
game_type	6
launch_speed_angle	6
bb_type	4
n_thruorder_pitcher	4
type	3
if_fielding_alignment	3
of_fielding_alignment	3
woba_denom	2
babip_value	2
stand	2
p_throws	2

Table 2: Unique Count for Categorical Columns

Does the dataset align with the project's objectives? Does it have the necessary features for the intended analysis or modeling?

This dataset is the premier datasets for tracking in-game baseball data. It has been used by professionals and amateurs alike to examine player performance and batted ball outcomes. It contains numerous metrics and statistics that should be more than necessary for our modeling.

Scalability:

Is the dataset size manageable with the available resources? Does the dataset require advanced techniques to handle its size (e.g., distributed processing)?

While the full dataset is fairly large (over 5 million rows of data), we are only using 2024 data as training data. This one year of data has 757,713 rows of data and is under 500 MB in when stored as a .csv file. Any reasonably modern computer should be able to process this amount of data, and therefore no advanced techniques (distributed processing, making use of high-powered computer clusters) are needed.

Transformations:

Does the data need preprocessing, such as normalization, standardization, or scaling? Are there opportunities to create new features that could improve the model or analysis?

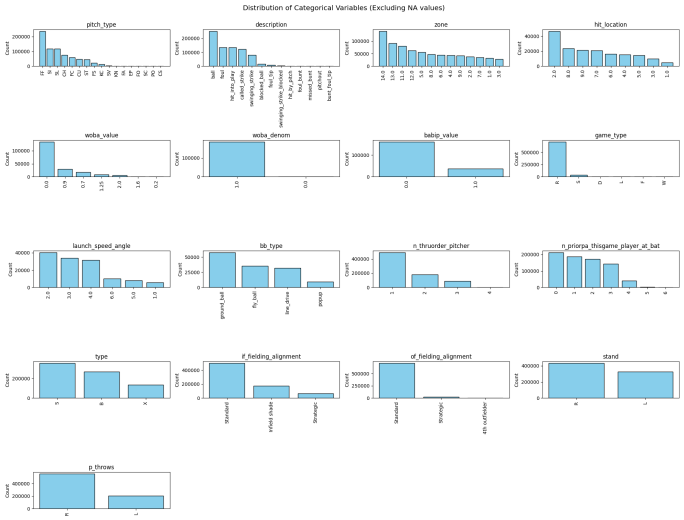

Figure 1: Distribution of Categorical Variables

The data from Statcast is of extremely high quality, so no large errors should be present in the data. Continuous numeric data such as 'ax' (acceleration in the x-dimension), 'release_pos_z' (Vertical Release Position), 'launch_speed' (Exit velocity), and 'hit_distance_sc' (Projected hit distance) can be normalized to ensure consistent values for comparison and avoid accidentally weighting some values more than others in certain models.

Data Encoding:

How should categorical variables be encoded (e.g., one-hot, label encoding)? Are there any temporal or sequential features that require specific transformations?

The majority of the categorical variables have a fairly limited number of unique classes (the majority having 6 or fewer classes, with the maximum number of classes being 17 for 'pitch_type'). These variables can be one-hot encoded. The variables 'stand' and 'p_throws' are binary strings, so these can just be encoded as binary values. No temporal transformations are needed.

Predictive Power:

Do the features contain sufficient predictive information for the target variable? Is feature selection or dimensionality reduction necessary?

The features definitely contain sufficient predictive information for the target variable. Many supervised learning projects have been done using these Statcast metrics, the tougher part will be the feature selection and we will likely have to

do some dimensionality reduction. There are likely dozens of features we will consider using in the model so there is work to be done for choosing the right features.

Target Variable:

If this is a supervised learning problem, what is the target variable, and is it well-defined? Is the target variable balanced, or does it require special handling (e.g., resampling)?

It is a supervised learning problem with the target variable being batted ball metrics, mainly exit velocity along with launch and spray angle. Based on the post-pitch metrics, we can use existing wOBA models to gauge just how productive or unproductive the pitch would be. The target variable I expect would be relatively balanced, but we may need to do some resampling if needed.

Validation Strategy:

How will you split the dataset for training, validation, and testing? Are there temporal or spatial dependencies that need to be preserved during splitting?

We are able to split 2024 data into training and validation groups, and then can use previous years (just 2023 should suffice) for testing data. There shouldn't be issues with temporal or spatial dependencies given the nature of what we are trying to predict.

Data Leakage:

Are there any risks of data leakage, where information from the test set inadvertently influences the model during training?

We will need to use known batted ball outcomes from 2024 data to train the model, but information from the test set should not inadvertently influence the training. Once the model is trained on a large chunk of data from 2024, we will test it on other unseen pitch data and see if it is producing likely outcomes that are similar to the actual post-pitch result.

Interpretability:

Can the dataset and analysis provide interpretable insights for stakeholders? How will the results be communicated (e.g., visualizations, metrics)?

The dataset and analysis should provide interpretable insights for stakeholders, including feature importance rankings and SHAP values. We will communicate the results both with metrics and visualizations such as kernel density plots,

noting which type of pitch profile we think will produce the most pitcher-friendly result.

Limitations:

What are the limitations of this dataset for the current project? What additional data would enhance the analysis?

The data is very thorough based on what the radar and camera technology is able to pick up, but there are always some potential errors when it comes to getting the most accurate metrics. The introduction of Hawkeye a few years ago has certainly improved the data collection from 2017 when Statcast was introduced, and the numbers are generally very accurate, but every once in a while there may be a misread with the radars. Some additional data that captures more context of the game state such as environmental data or park factors could help, especially when looking at flyballs.