Proceedings of the CONF-MLA 2026 Symposium: Explainable Computing, Modeling & Data Science in Complex Systems
DOI: 10.54254/2755-2721/2026.GF36448

Quantifying Pitcher Dominance: A Machine Learning Approach to Predicting Swinging Strikes from Statcast Pitching Data

Ruicheng Sha

Saint James School, Hagerstown, USA
richardsrc3080@outlook.com

Abstract. In modern baseball, the relative contributions of velocity, spin, and pitch location to swinging-strike outcomes are still not well understood, and many earlier studies mix up location-driven outcomes with actual pitch quality. This study uses MLB Statcast pitch-level data from 2024–2025, drawn from 221,982 pitches thrown by World Baseball Classic 2026 roster pitchers across 16 countries, to analyze what physical factors predict pitch effectiveness. To separate pitch quality from location effects, this study defines a zone-controlled binary target: in-zone swinging strikes versus in-zone hard-hit contact, while excluding out-of-zone pitches, called strikes, foul balls, and other ambiguous outcomes. This yields a modeling sample of 24,710 pitches with a balanced 45.0%/55.0% class split that needs no resampling. Using a stratified 70/30 split that preserves this class ratio, three classification models are trained and compared: logistic regression, decision tree and random forest, with the two tree models tuned by cross-validated grid search. Exploratory data analysis shows that pitch velocity and vertical plate location separate the two classes most clearly. The random forest classifier achieves the best performance with an accuracy of 67.84%, an ROC AUC of 0.7359, and an F1-score of 0.62, outperforming the decision tree (accuracy 65.72%, AUC 0.7042) and the logistic regression baseline (accuracy 57.37%, AUC 0.5759). Notably, feature importance analysis shows that stuff-related features together account for 46.7% of importance in the random forest—nearly as much as the two location features (53.3%)—showing that pitch quality, not just placement, is an important and often overlooked driver of swinging strikes.

Keywords: swinging strike, pitch effectiveness, Statcast, random forest, feature importance

1. Introduction

Since MLB introduced the Statcast system in 2015, every pitch at the major-league level is recorded with high precision, capturing release speed, spin rate, horizontal and vertical break, and the exact location where the ball crosses home plate [1]. Among all possible pitch out-comes, the swinging strike is one of the most direct signals of pitcher dominance: the batter completely failed to make contact. Understanding what physical and spatial characteristics lead to swinging strikes is very important for pitch design, game-calling strategy, and pitcher development.

DOI: 10.54254/2755-2721/2026.GF36448

Despite the growing availability of Statcast data, the relative importance of velocity, spin, break, and location in producing swinging strikes is still not well understood. A key problem in earlier studies is that their swinging-strike targets often include outcomes heavily influenced by location, such as foul tips and out-of-zone chases, making location features look more important than they really are and hiding the contribution of pitch quality.

This study addresses this problem by using a zone-controlled target definition that separates pitch quality from location effects. We only look at pitches *within the strike zone* and compare in-zone swinging strikes (where pitch quality beat the batter) against in-zone hard-hit contact (where the batter hit the ball hard). This design ensures both classes represent pitches in hittable locations, revealing pitch quality's role beyond just location.

The research questions are: (1) Which pitch features are most predictive of swinging-strike outcomes when location effects are controlled? (2) How do velocity, spin, and break (collectively, "stuff") compare to pitch location in determining pitch effectiveness? To answer these questions, this paper compares three models of increasing complexity—logistic regression, decision tree, and random forest—and evaluates them using accuracy, precision, recall, F1-score, and ROC AUC. The overall research pipeline is summarized in Figure 1.

2. Related work

Pitch-tracking data introduced by PITCHf/x and later Statcast enabled precise measurement of pitch trajectories [2], spurring a large body of work on pitch classification and characterization. Studies have improved raw pitch-type labels through statistical learning [3], applied multi-class classifiers to identify pitch types from velocity and movement [4], and used clustering to find pitch subtypes and effective pairings [5]. A related strand models the batter side of the interaction, including plate-discipline and swing-decision modeling [6], swing-tracking metrics [7], and cue-based pitch-type anticipation [8].

A more directly related line of work predicts swinging-strike outcomes from pitch characteristics. Logistic regression and random forests have been used to predict swinging strikes, framing whiff-inducing ability as a marker of pitcher value [9, 10]. A common limitation is that their target includes any swinging strike—out-of-zone chases and foul tips included. Mixing these location-driven outcomes with genuine quality-driven whiffs tends to inflate the apparent importance of location features and obscure the contribution of velocity, spin, and movement. This work adds to this literature by (1) introducing a zone-controlled target definition that separates pitch quality from location, (2) comparing three model families from a linear baseline to a tuned ensemble, and (3) showing that stuff-related features account for nearly half of the predictive power when location effects are properly controlled.

3. Dataset

3.1. Study overview

Figure 1 summarizes the overall workflow of this study, from the raw Statcast feed through the zone-controlled target definition and stratified split to the training and comparison of the three models.

DOI: 10.54254/2755-2721/2026.GF36448

3.2. Data source

The data used in this study comes from the WBC 2026 Scouting Dataset, publicly available on Kaggle. The dataset was collected from Baseball Savant using the pybaseball Python library and contains MLB regular season Statcast data for players on the WBC 2026 (World Baseball Classic) rosters. The raw pitching file (statcast_pitchers.csv) contains 221,982 pitch-level observations covering 16 countries, with game dates ranging from March 20, 2024 to November 1, 2025.

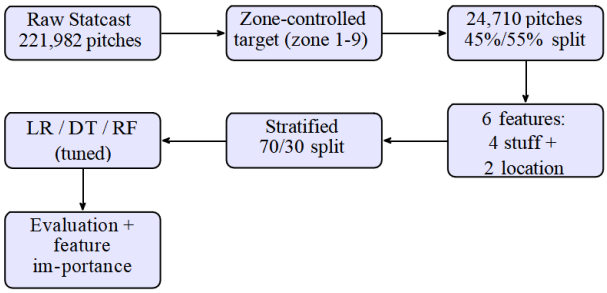

Figure 1. Research pipeline: from raw Statcast data to model comparison and feature-importance analysis

3.3. Variable description

The dataset contains 28 columns. Table 1 summarizes the six features selected for modeling.

Table 1. Description of selected variables

Variable	Description	Type	Range
release_speed	Pitch velocity at release (mph)	Continuous numeric	54.9–102.5
release_spin_rate	Spin rate at release (rpm)	Continuous numeric	91–3482
pfx_x	Horizontal break (feet)	Continuous numeric	−2.07–2.06
pfx_z	Vertical break (feet)	Continuous numeric	−1.76–1.99
plate_x	Horizontal location at plate (feet)	Continuous numeric	−0.83–0.85
plate_z	Vertical location at plate (feet)	Continuous numeric	1.16–4.12

Note: All six features are continuous numeric variables. The two location features deserve ex-plicit definition, since they are central to our research question: plate_x is the ball's horizontal position as it crosses home plate (in feet, catcher's perspective, 0 at plate center), and plate_z is its vertical position (in feet above the ground). The reported range of plate_x and plate_z is for the zone-controlled subset (zone 1-9 only), not the full dataset. The remaining four are stuff features describing how the pitch moves.

3.4. Target variable

To separate pitch quality from location, we define a zone-controlled target using only pitches within the strike zone (Statcast zones 1-9): positive class=in-zone swinging strikes (whiffs on hittable pitches), negative class=in-zone hard-hit contact (exit velocity≥95 mph). Excluded outcomes include out-of-zone pitches, called strikes, foul balls, and weak contact. This design ensures both classes represent hittable pitches, isolating pitch quality from location effects.

DOI: 10.54254/2755-2721/2026.GF36448

3.5. Class distribution

After removing rows with missing values (0.4%), the modeling sample contains 24,710 observations with a 45.0%/55.0% class split (Table 2), requiring no resampling.

Table 2. Class distribution of the modeling sample

Class	Count	Proportion
In-Zone Whiff (positive)	11,128	45.0%
In-Zone Hard Contact (negative)	13,582	55.0%

4. Exploratory data analysis

This study quantifies feature separation between in-zone whiffs and hard contact using Cohen's d (Table 3). Only three features show visible separation: plate_z (d = 0.150), release_speed (d = 0.114), and plate_x (d = 0.087). Vertical location separates the classes most clearly (Figure 2): whiffs cluster at zone edges while hard contact concentrates in the middle. Whiff-inducing pitches are slightly slower (89.8 vs. 90.5 mph, Figure 3), consistent with off speed deception. The remaining features show near-zero univariate separation but contribute through interactions in ensemble models.

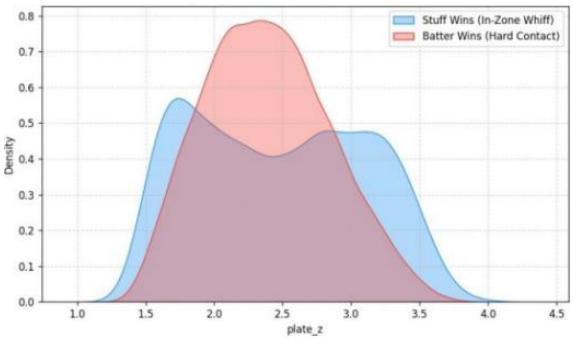

Figure 2. Distribution of plate_z(vertical location) by pitch outcome

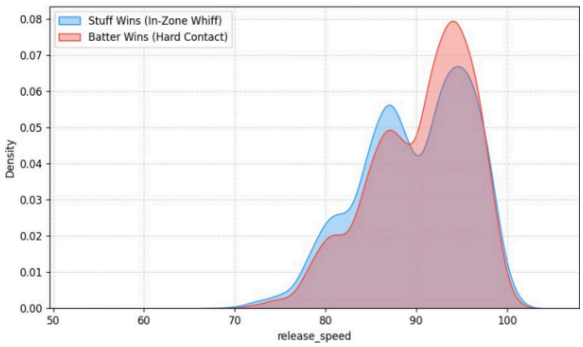

Figure 3. Distribution of release_speedby pitch outcome

Table 3. Univariate class separation by feature (Cohen's d, absolute value)

Feature	Cohen's d	Interpretation
plate_z	0.150	negligible
release_speed	0.114	negligible
plate_x	0.087	negligible
pfx_x	0.060	negligible
pfx_z	0.040	negligible
release_spin_rate	0.010	negligible

Note: $|d| < 0.2$ negligible, $0.2 \leq |d| < 0.5$ small, $0.5 \leq |d| < 0.8$ medium, $|d| \geq 0.8$ large.

DOI: 10.54254/2755-2721/2026.GF36448

4.1. Feature correlations

Figure 4 shows the Pearson correlation heatmap. The strongest relationship is a positive correlation between release_speed and pfx_z (faster pitches carry more vertical break), while pfx_x correlates negatively with release_speed (slower curveballs and changeups break more horizontally). The two location features are only weakly correlated, so horizontal and vertical commands are largely independent.

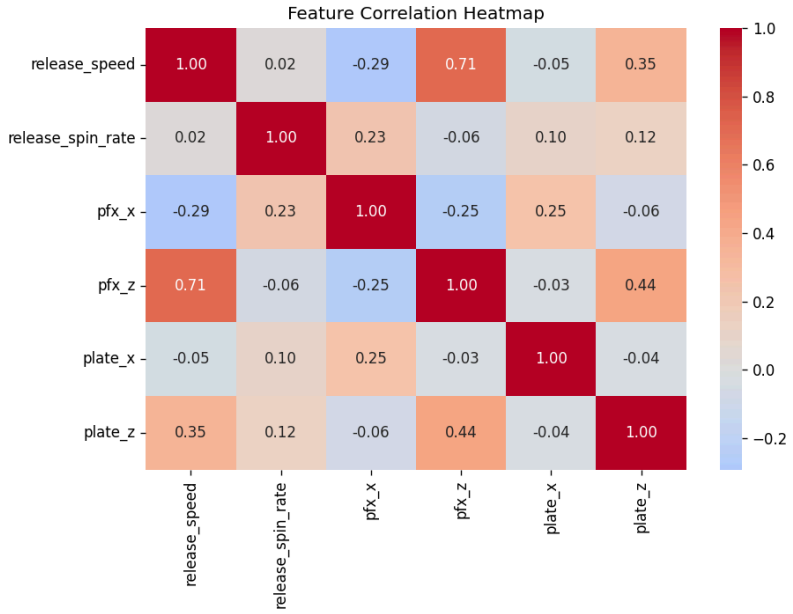

Figure 4. Pearson correlation heatmap of selected features

5. Data preprocessing

Features are Z-score standardized for logistic regression, tree models are scale-invariant and require no standardization. The sample is split 70/30 (training/test) with stratified sampling to preserve the 45:55 class ratio, yielding 17,297 training and 7,413 test observations.

6. Modeling

6.1. Model selection rationale

Two tree-based models of increasing complexity are compared: (1) Decision tree, a non-linear model that captures feature interactions through hierarchical splits, and (2) Random forest, an ensemble method that combines many decision trees to reduce variance and improve generalization.

6.2. Decision tree algorithm

A decision tree builds a hierarchical structure by recursively partitioning the feature space. At each node, the algorithm selects the feature and threshold that maximize purity of the resulting subsets. We use Gini impurity as the splitting criterion:

$$\text{Gini}(D) = 1 - \sum_{k=1}^{k} p_k^2 \quad (1)$$

DOI: 10.54254/2755-2721/2026.GF36448

where pk is the proportion of class k in dataset D. The tree grows until stopping conditions are met (maximum depth, minimum samples per leaf, or pure nodes).

6.3. Random forest algorithm

The random forest is a Bagging (Bootstrap Aggregating) ensemble method. Bagging trains multiple base learners independently on bootstrap samples and combines predictions by voting, primarily reducing variance. This contrasts with Boosting, which trains learners sequentially to correct predecessors' errors, reducing bias. The random forest adds randomness at two levels: each tree trains on a bootstrap sample (*row-level*), and each split considers only a random feature subset (*column-level*). This double randomization decorrelates trees, driving variance reduction.

6.4. Hyperparameter tuning

Both tree models are tuned by 5-fold cross-validated grid search on the training set, using ROC AUC as the scoring metric. For the decision tree, the best configuration is max_depth=10, min_samples_leaf=50, min_samples_split=10, with cross-validated AUC of 0.7048.

The random forest uses a staged tuning strategy: first optimizing the number of trees (n_estimators), then tree complexity parameters, and finally row/column sampling rates. The final configuration is n_estimators=300, max_depth=12, min_samples_leaf=10, min_samples_split=2, max_features=sqrt, max_samples=0.6, achieving cross-validated AUC of 0.7388. Table 4 summarizes the tuning results.

Table 4. Cross-validated tuning summary (mean 5-fold ROC AUC on the training set)

Model / stage	Best configuration	CV AUC
Decision Tree (grid)	depth 10, leaf 50, split 10	0.7048
RF stage 1 (trees)	n_estimators 300	0.7304
RF stage 2 (complexity)	depth 12, leaf 10, split 2	0.7388
RF stage 3 (sampling)	max_features sqrt, max_samples 0.6	0.7388

7. Results

7.1. Model comparison

Table 5 shows the evaluation metrics for all three models on the test set (7,413 observations). The Receiver Operating Characteristic Area Under the Curve (ROC AUC) is reported alongside the threshold-based metrics, the random forest's out-of-bag (OOB) score is reported separately in its own subsection because it is an internal metric specific to bagging.

Table 5. Model performance comparison on the test set (7,413 observations)

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	0.5737	0.5500	0.2900	0.3800	0.5759
Decision Tree	0.6572	0.6400	0.5400	0.5900	0.7042
Random Forest	0.6784	0.6700	0.5700	0.6200	0.7359

DOI: 10.54254/2755-2721/2026.GF36448

The random forest achieves the best performance on every metric. The logistic regression baseline is weakest: its low recall (0.29) reflects an over-conservative decision boundary. The AUC jump from 0.58 to 0.74 indicates non-linear feature interactions.

7.2. Confusion matrices

Tables 6–8 show confusion matrices (rows = actual, columns = predicted, class 0 = hard contact, class 1 = whiff). Logistic regression's large false-negative count (2,368) reflects its conservative strategy. The random forest reduces false negatives to 1,428, improving recall and F1.

Table 6. Logistic regression confusion matrix

		Pred 0	Pred 1
Actual	0	3,283	792
	1	2,368	970

Table 7. Decision tree confusion matrix

		Pred 0	Pred 1
Actual	0	3,077	998
	1	1,543	1,795

Table 8. Random forest confusion matrix

		Pred 0	Pred 1
Actual	0	3,119	956
	1	1,428	1,910

7.3. Feature importance analysis

The random forest assigns: plate_z 37.9%, plate_x 15.4%, release_speed 14.0%, pfx_z 11.9%, release_spin_rate 10.7%, pfx_x 10.1%. Grouping by category: location features (53.3%) versus stuff features (46.7%)—a nearly balanced split that contrasts with uncontrolled targets where location typically dominates (70%+).

8. Conclusion

This study analyzed MLB Statcast data from WBC 2026 roster pitchers to identify the physical and spatial factors that contribute to swinging-strike outcomes. By using a zone-controlled target definition that isolates pitches within the strike zone, we separated pitch quality effects from location-driven outcomes and compared three classification models: logistic regression, decision tree, and random forest. The random forest achieved the best performance with an accuracy of 67.84%, ROC AUC of 0.7359, and F1-score of 0.62.

The key finding is that pitch quality accounts for nearly half (46.7%) of the Random Forest's predictive power, with location features contributing 53.3%. This balanced split challenges earlier

DOI: 10.54254/2755-2721/2026.GF36448

studies where location typically dominated (over 70% importance), and confirms that stuff-related features—velocity, spin rate, and movement—are critical drivers of swinging strikes once location is controlled. Within the strike zone, whiffs are associated with slightly slower velocity (89.8 vs. 90.5 mph) and greater vertical break, indicating that offspeed deception and sharp movement drive in-zone whiffs more effectively than raw velocity. The large performance gap between logistic regression (AUC 0.58) and tree-based models (AUC 0.70–0.74) demonstrates that the relationship between pitch features and outcomes involves substantial feature interactions that a linear model cannot capture.

These findings have practical implications: pitcher development programs should invest in pitch quality—spin, movement design, and offspeed deception—alongside command training, and game-calling strategies should exploit high-quality offspeed pitches in swing-likely counts. This study has several limitations. The dataset covers only WBC 2026 roster pitchers during 2024-2025, omits contextual features such as count situation, pitch sequence, and batter handedness, and compares only three model families. Future work should expand to all MLB pitchers across multiple seasons, incorporate batter-specific characteristics and situational con-text, and test advanced models such as gradient boosting or neural networks to achieve more accurate and actionable matchup-level predictions.

References

[1] Kagan, D., & Nathan, A. M. (2017). Statcast and the baseball trajectory calculator. *The Physics Teacher, 55*(3), 134–136.
[2] Fast, M. (2010). What the heck is PITCHf/x? *The Hardball Times Annual 2010*. https://tht.fangraphs.com/tht-live/pitchf-x-reference/
[3] Pane, M. A., Ventura, S. L., Steorts, R. C., & Thomas, A. C. (2013). Trouble with the curve: Improving MLB pitch classification. *arXiv preprint arXiv:1304.1756*.
[4] Sidle, G., & Tran, H. (2018). Using multi-class classification methods to predict baseball pitch types. *Journal of Sports Analytics, 4*(1), 85–93.
[5] Society for American Baseball Research. (2020). Using clustering to find pitch sub-types and effective pairings. *SABR Journal*. https://sabr.org/journal/article/using-clustering-to-find-pitch-subtypes-and-effective-pairings/
[6] Yee, R., & Deshpande, S. K. (2023). Evaluating plate discipline in Major League Baseball with Bayesian additive regression trees. *arXiv preprint arXiv:2305.05752*.
[7] Powers, S., & Yurko, R. (2025). Interpreting variation in baseball swing tracking metrics. *arXiv preprint arXiv:2507.01238*.
[8] Takamido, R., Suzuki, C., & Nakamoto, H. (2026). A data-driven analysis of spatiotemporal cues and experience accumulation effects for pitch type prediction. *PLOS ONE*.
[9] Cheng, R. (2016). Hit or miss: Attempting to predict swinging strikes. *The Hardball Times*, May 12. https://www.fangraphs.com/tht/hit-or-miss-attempting-to-predict-swinging-strikes/
[10] The Hardball Times. (2018). Using movement, velocity & location to predict a swinging strike. *The Hardball Times Annual 2018*. https://tht.fangraphs.com/tht-annual-2018/using-movement-velocity-location-to-predict-a-swinging-strike/