Uncovering Patterns in MLB Performance Data

Nevena Ciganovic, Adele Lauzon, & Joanna Lo

Contributions

Nevena Ciganovic: Methods & Results (RQ1), Discussion

Adele Lauzon: Methods & Results (RQ2, RQ3), Discussion

Joanna Lo: Introduction, Data Description, Discussion

1. Introduction

Major League Baseball (MLB) has long been at the forefront of sports analytics, with teams increasingly relying on data to optimize player performance, refine strategies, and gain a competitive edge. The introduction of Statcast technology in 2015 revolutionized the field by enabling precise, high-dimensional tracking of pitch movement, swing mechanics, and batted ball characteristics. These data offer a unique opportunity to move beyond traditional box-score statistics and instead model the processes that drive game outcomes—from how a pitch is thrown to where a ball lands.

Existing work has leveraged Statcast data to build models that predict the outcome of a batted ball using features like exit velocity and launch angle. For example, MLB's own expected batting average (xBA) and expected wOBA (xwOBA) metrics are derived from such models, typically using nearest-neighbour or regression-based approaches to estimate the likely results based on contact quality (Sharpe, 2017; Albert 2018). These metrics have proven useful in isolating hitter skill from external factors like defense or ballpark effects (Arthur, 2016). However, most existing approaches rely on a narrow set of features, often excluding richer pitch-level or swing-context variables that may hold additional predictive value. Furthermore, violations of modeling assumptions like linearity and independence in traditional regression-based models motivate the use of more flexible machine learning algorithms (Petti, 2016).

Although the spatial location of a batted ball is known to be highly predictive of its classification, relatively few studies have explored unsupervised learning methods to discover natural groupings in hit-type data. Much of the literature relies on manually labeled events or predefined launch angle bands to distinguish fly balls, line drives, or grounders (Daley-Harris, 2016). Yet research shows that hit trajectories are often non-convex and unevenly distributed, suggesting that algorithms like spectral clustering may offer a more appropriate way to uncover structure in hit-location data (Perpetua, 2017). However, such applications remain sparse, and few models incorporate pixel-level hit coordinate data to distinguish between nuanced outcomes like singles, doubles, or sacrificial hits.

In addition, many studies have emphasized raw performance metrics over in-game context, leaving open questions about how factors like pitch type, home/away status, or inning number affect

player outcomes. While some internal team analyses may incorporate these situational features, few published studies evaluate their importance systematically, and traditional linear models may be insufficient to handle the complex, high-dimensional feature space involved (Mann, 2024). Moreover, while PCA has been widely used in sports analytics for dimensionality reduction, its combination with supervised learning techniques to isolate key situational features remains relatively unexplored in baseball.

To address these gaps, this project applies various statistical and machine learning techniques to uncover patterns in MLB performance data. Specifically, we focus on three research questions:

- **RQ1:** Can we predict whether a batted ball will result in a hit, out, or home run based on swing characteristics?
- **RQ2:** Can we identify what type of ball was hit (bunt, single, home run, etc.) based on the hit coordinates of the batted ball?
- **RQ3:** What are the key factors contributing to player performance in different game situations?

Each question is grounded in open problems within baseball analytics. The goal is to build models that not only predict outcomes, but also offer insight that could potentially help players, coaches, and analysts make better decisions in the field.

2. Data Description

The dataset used in this project is the MLB Statcast Data, a publicly available dataset on Kaggle that provides detailed player statistics collected using high-speed cameras and radar technology (Kaggle, 2022). It captures a wide range of performance metrics across multiple MLB seasons, making it an ideal resource for analyzing trends in player behaviour and skill development. Although the full dataset spans give CSV files covering the 2017 to 2021 seasons, we focus exclusively on the 2019 season to maintain consistency in the data collection technology and limit computational overhead. The 2019 file contains a total of 732,473 observations, with each observation representing an individual pitch or batted ball event. A total of 93 features—both continuous and categorical—quantify various aspects of pitching, batting, and fielding, enabling a comprehensive statistical analysis of player performance.

Minimal data preprocessing was required. Deprecated features, as well as temporal variables and player IDs not relevant to our chosen research questions (e.g., date of the game), were removed. Our research questions only relate to events where the ball was pitched successfully, so any instances where the release speed of the pitch was missing were removed. Finally, a variable indicating the type of game only took one value ("R" for regular season), and therefore was also removed.

Three new features were engineered. First, a variable *team_at_bat* was created to indicate whether the home team or away team was at bat during the time of the event. This information was gleaned from whether it was the top or the bottom of the inning. Second, a variable *type_of_hit* was created based on the events that occurred as a result of the hit (gleaned from the *events* column). For example, if the resulting event was described as a sacrificial fly with a double play, or a sacrificial fly, *type_of_hit* took the value "sac_fly." This new variable took six values representing the following outcomes: sacrificial fly, sacrificial bunt, home run, single, double, and triple. Third, a variable *outcome* was created by categorizing each play into one of three classes: "home_run," "hit," or "out." The original *events* column was recorded such that "home_run" remained as is, all hit events ("single," "double," and "triple") were grouped under "hit," and all types of outs (e.g., "field_out," "force_out," "fly_out") were grouped under "out." Any rows with events not falling into these categories were removed, and the *outcome* variable was converted to a factor to facilitate classification modeling.

To provide an overview of the numeric data, we computed summary statistics for several variables relevant to our research questions: *launch_speed*, *launch_angle*, *release_speed*, *plate_x*, *plate_z*, *hit_distance_sc*, and *woba_value*. These features were selected because they present key components of swing quality, pitch placement, batted ball distance, and overall player performance. The summary statistics (Figure 1) reveal substantial missingness in many of these variables, which motivated the preprocessing steps described below.

```
exit_velocity     launch_angle    release_speed       plate_x           plate_z       hit_distance_sc    woba_value
Min.   :  7.6    Min.   :-89.0    Min.   : 50.60   Min.   :-5.03000   Min.   :-4.130   Min.   :  0.0    Min.   :0.0
1st Qu.: 73.9    1st Qu.: -3.0    1st Qu.: 84.50   1st Qu.:-0.55000   1st Qu.: 1.630   1st Qu.: 30.0    1st Qu.:0.0
Median : 82.9    Median : 18.0    Median : 89.80   Median : 0.04000   Median : 2.250   Median :173.0    Median :0.0
Mean   : 83.9    Mean   : 16.6    Mean   : 88.68   Mean   : 0.03845   Mean   : 2.246   Mean   :165.2    Mean   :0.3
3rd Qu.: 95.4    3rd Qu.: 37.0    3rd Qu.: 93.40   3rd Qu.: 0.62000   3rd Qu.: 2.870   3rd Qu.:260.0    3rd Qu.:0.7
Max.   :125.3    Max.   : 89.0    Max.   :104.30   Max.   : 6.29000   Max.   :12.210   Max.   :526.0    Max.   :2.0
NA's   :524591   NA's   :524591                    NA's   :21         NA's   :21       NA's   :533100   NA's   :540998
```

Figure 1. Summary statistics for selected numeric variables.

We also generated three initial exploratory visualizations aligned with each of our three research questions. For RQ1, we plotted the distribution of exit velocity (*launch_speed*) across batted ball outcomes (Figure 2). As expected, home runs tended to have the highest median exit velocity, followed by hits and outs. The visualization supports the hypothesis that exit velocity is a useful predictor of batted ball outcomes. For RQ2, we visualized the spatial distribution of hit coordinates by *type_of_hit* (Figure 3), which revealed clear spatial clustering by hit type. For example, sacrificial bunts are concentrated near home plate, while singles and doubles extend further into the field. Finally, for RQ3, we plotted *woba_value* by pitch type (Figure 4), highlighting variability in offensive performance across different pitch types.

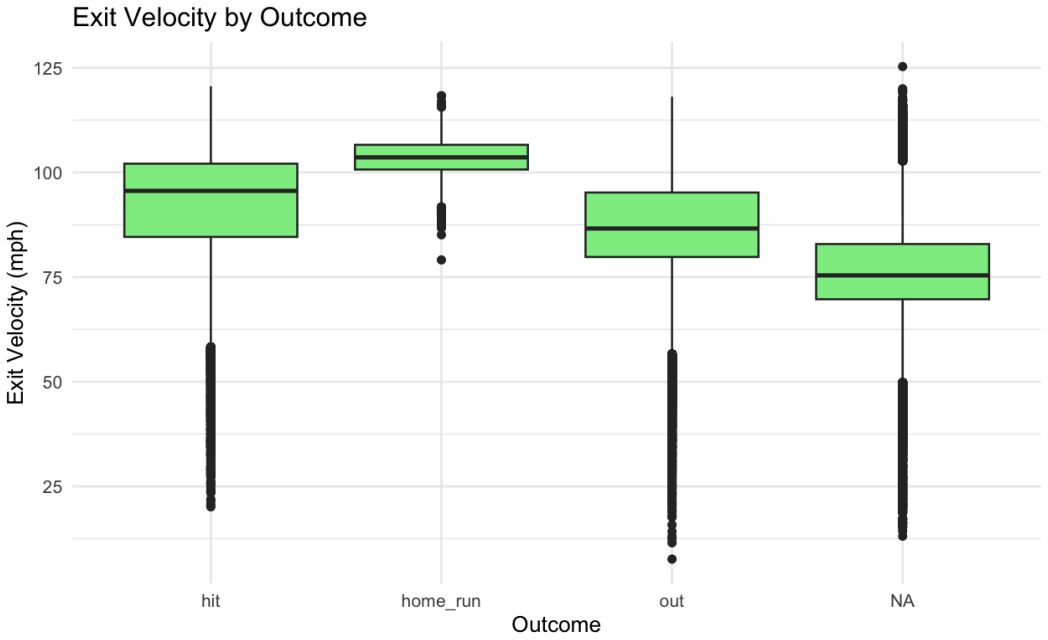

Figure 2. Exit velocity by batted ball outcome.

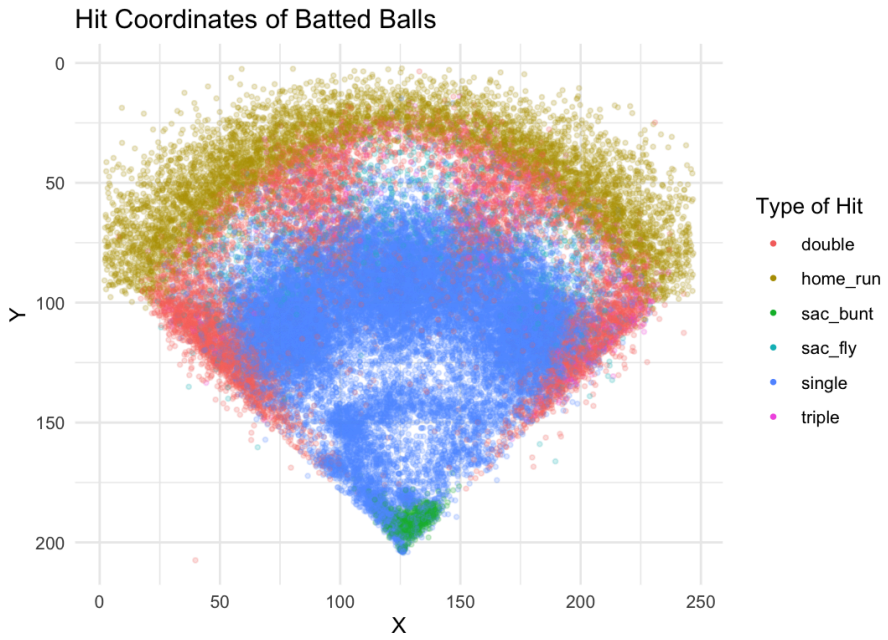

Figure 3. Hit location colored by type of hit.

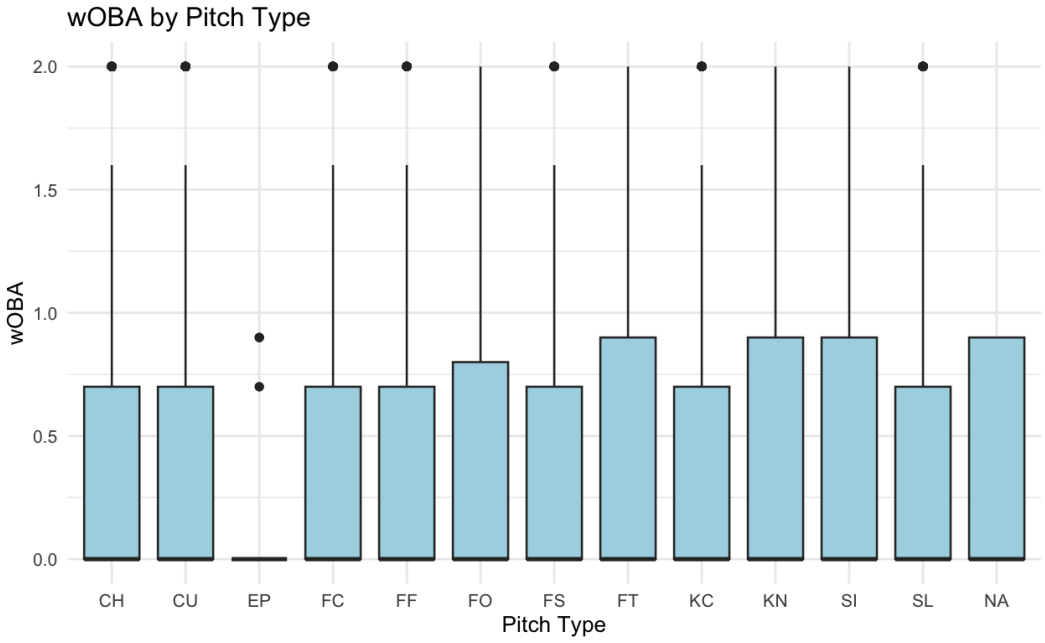

Figure 4. wOBA values across different pitch types.

Additional preprocessing steps were taken depending on the research question. For RQ1, to prepare the dataset for modeling, we selected only the relevant swing-related features and ensured data completeness. Specifically, a subset of swing characteristics predictive of the outcome—such as launch

angle, launch speed, pitch type, and plate location—were extracted from the cleaned dataset using a curated list of "selected_features." In addition, all rows containing missing (NA) values in any of the selected columns were removed. This ensured that the modeling dataset consisted of only fully observed, relevant features, allowing for consistent and reliable training and evaluation of classification models.

For RQ2, we were only concerned with instances where the ball was actually hit by the batter and therefore had non-missing hit coordinates. As such, we removed any rows with missing hit location data. Further, since spectral clustering is computationally expensive, we randomly sampled 5,000 rows to conduct the clustering analysis.

For RQ3, the dataset was reduced to include only variables that are fixed at the time of bat. For all remaining columns, it was found that in cases of numeric variables, imputing 0 for missing values was appropriate and did not distort meaning. For example, if *launch_speed* is missing, this indicates that the pitch was not hit, and imputing a 0 correctly reflects this. Similarly, "None" was imputed for character variables with missing values, a strategy that was found to be valid for all columns included in the RQ3 analysis.

3. Methodology

3.1. Research Question 1

To assess whether multinomial logistic regression would be suitable in predicting outcome (hit, out, home run) of a batted ball based on swing characteristics, we examined two key assumptions: multicollinearity and linearity of the log odds. A correlation plot (Figure 5) of the numeric predictors revealed several clusters of highly correlated variables, indicating potential multicollinearity, which can distort the interpretation of model coefficients. Additionally, the assumption of linearity between continuous predictors and the log odds of the outcome categories was evaluated using density plots (Figure 6) grouped by outcome. These plots demonstrated that for several variables, the relationship between predictor values and the log odds was clearly non-linear.

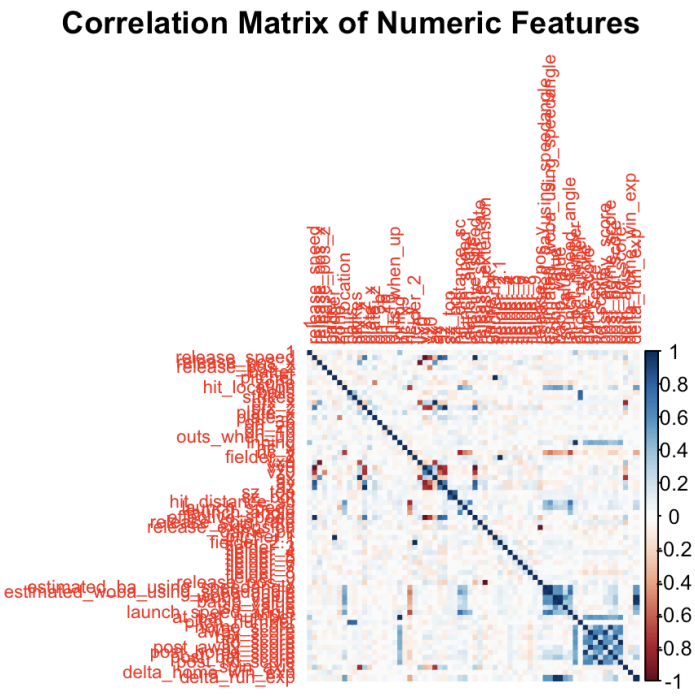

Figure 5. Correlation matrix of numeric features.

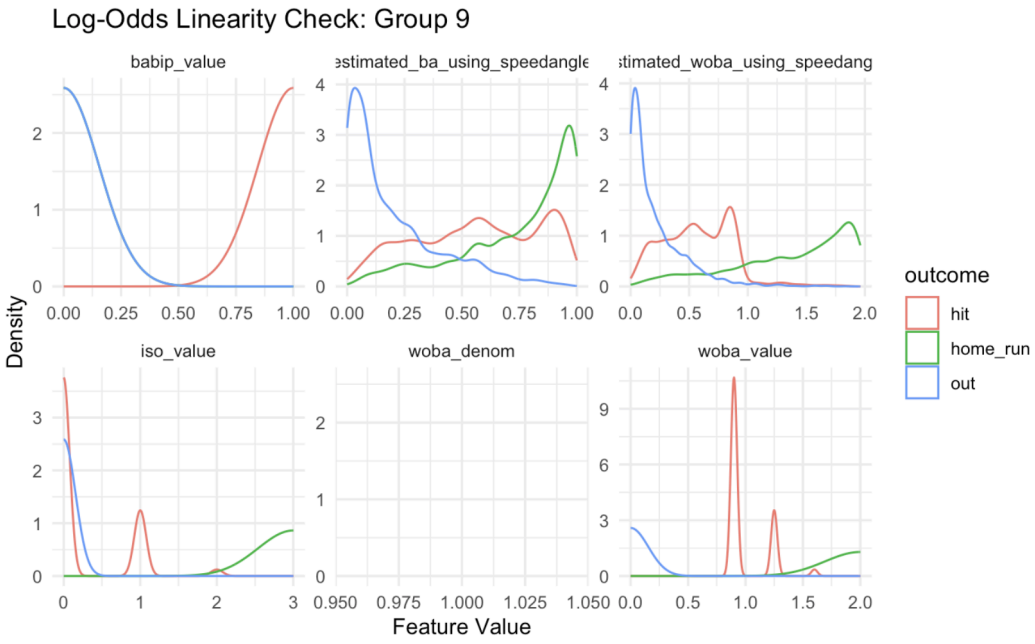

Figure 6. Log-odds linearity check for a sample of features.

Next, to assess whether LDA or QDA would be suitable in predicting outcome, we examined a key assumption: predictors are normally distributed within each class. To assess normality, the Shapiro-Wilk test was applied to numeric features within each outcome class. While this test can be

sensitive to large sample sizes, it flagged several features with low p-values, suggesting potential deviations from normality. Additionally, a density plot of the sample feature "launch_speed" (Figure 7) shows clear differences in distribution across classes. These results indicate that normality may not be fully satisfied, which could impact LDA/QDA performance. Given violation of assumptions in these three models, logistic regression, LDA and QDA are not appropriate modeling approaches for this data, motivating the use of more flexible, non-parametric machine learning models instead.

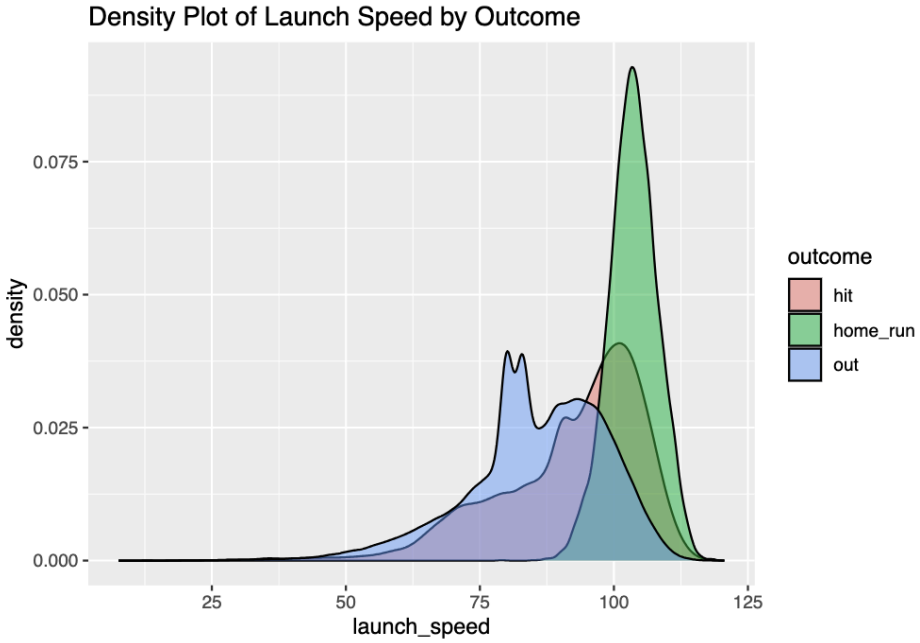

Figure 7. Density plot for a sample feature.

Hence, to classify baseball swing outcomes as either hit, home run, or out, we implemented three supervised machine learning models: Random Forest, XGBoost, and a Feedforward Neural Network. These models were chosen to explore different statistical learning methods — ensemble trees, gradient boosting, and neural architectures — offering complementary insights into the predictive power of swing characteristics. An 80/20 train-test split stratified by outcome ensured balanced class representation. All models were trained using 5-fold cross-validation and evaluated on the same hold-out test set.

We first implemented a Random Forest classifier due to its flexibility, robustness to multicollinearity, and ability to model complex, non-linear relationships. This model is an ensemble of

decision trees that uses bagging and random feature sampling to reduce overfitting and variance. The model was trained with 200 trees, using Gini impurity as the splitting criterion. Feature importance was extracted from the trained model to identify influential swing variables. Overall, this model served as a strong baseline model due to its interpretability and ability to handle high-dimensional, correlated, and non-linearly separable data without requiring extensive feature engineering or transformation.

Next, we implemented an XGBoost classifier due to its ability to model non-linear relationships and handle multicollinearity, both of which were present in the dataset. XGBoost is a powerful gradient boosting framework known for its speed and performance on structured data. Its built-in regularization helps prevent overfitting, and it performs well with imbalanced classes and high-dimensional numeric data, all of which is all present in our dataset. It builds additive decision trees sequentially, where each tree corrects errors from the previous one. This model is particularly well-suited to handling imbalanced classes and non-linear boundaries, which we found were common issues in our dataset during the exploratory data analysis.

Finally, we implemented a Feedforward Neural Network. Neural networks are well-suited for this task due to their ability to capture nonlinear relationships and complex interactions among predictors, which are common in gameplay data. A single-hidden-layer multilayer perceptron (MLP) was used to model non-linear relationships through a network of weighted connections. The tuning grid explored five combinations of the number of hidden nodes (size) and weight decay parameters to find the optimal configuration. The maximum number of iterations was increased to 200 to ensure the model had sufficient opportunity to converge. The architecture of this model is particularly useful when feature interactions are not known beforehand or are difficult to model explicitly. Moreover, neural networks are robust to correlated features and can generalize well with proper regularization and cross-validation.

Performance was assessed through multiple metrics including accuracy, precision, recall, and F1 score, ensuring a well-rounded understanding of each model's strengths and weaknesses across all outcome classes. Overall, this diversity in model structure allows for a more robust understanding of the predictive power and stability of swing characteristics in determining swing outcomes.

3.2. Research Question 2

To identify what type of ball was bat based on hit coordinates, we propose implementing a clustering algorithm to see if types of hit balls can be recovered based on location. Understanding likely outcomes based on where a ball lands in the field allows teams to develop offensive strategies targeted to that specific outcome. Preliminary EDA looking at the hit location of different types of hit balls indicated a complex data structure (Figure 3). Importantly, the clusters appear to be non-convex, especially home runs or balls that land deep in the outfield. Overall, clusters appear to have varying densities and are neither spherical nor elliptical in shape. Given this information, we chose to implement spectral clustering as our clustering algorithm.

Spectral clustering is known to be appropriate for non-convex, complicated data structures. However, the algorithm requires computing the pairwise similarities between all observations, which can be computationally intensive. Given our limitations, we elected to perform spectral clustering on a random sample of 5000 observations from the original dataset. After finding the weighted adjacency matrix (with neighbors = 20) and the Laplacian of this similarity matrix, we performed K-means (nstart = 10) on the Laplacian, with $K \epsilon [1, 6]$, where 6 is the total number of types of hit balls.

3.3. Research Question 3

To identify factors contributing to player performance in different game situations, we propose implementing Principal Component Analysis (PCA) and XGboost to see if specific features (if the batter is from home or away, what kind of pitch is thrown) impact weighted on-base average (wOBA). The feature set was reduced to variables that would be fixed at time-of-bat. For example, the type of pitch and the number of outs when the batter was up were retained, but the hit coordinates of the ball were removed. This resulted in 39 remaining features before one-hot encoding and 72 after encoding.

Understanding how variables fixed at time of bat impact player outcomes may inform training players ahead of certain game situations or certain teams. However, teams may not have the resources to target all potential variables in training. Therefore, it is in the best interest of the teams to narrow our focus to a small subset of variables shown to be most important in predicting wOBA. We first reduce the feature set by applying PCA, and creating a new set of features composed of the top

principal components that cumulatively capture ~90% of the original variance. Then, we fit XGboost 5-fold cross validation using this reduced dataset and rank the features by importance in predicting wOBA. XGboost was selected instead of traditional linear regression due to the large number of features and for the ability of the model to output a feature importance matrix. We then propose taking the five principal components deemed most important in predicting the outcome and investigating the loading vectors of these components to determine which of the original features are most relevant. Using this approach, we hope to identify 3-5 key variables that impact wOBA that teams can focus on in training.

4. Results

4.1. Research Question 1

The Random Forest model demonstrated strong overall performance in classifying swing outcomes, achieving a test set accuracy of 91.25%. The confusion matrix (Figure 8) shows the model was highly effective at predicting "out" outcomes, with a sensitivity of 96.2% and a precision of 91.6%. It also performed well on "home run" predictions, with 90.8% sensitivity and 86.5% precision. The "hit" class had slightly lower sensitivity (81.1%) but maintained strong precision at 91.5%, suggesting the model was more conservative when assigning this label. Finally, the model achieved strong macro-level classification metrics, with a precision of 0.899, recall of 0.894, and an F1 score of 0.895. These values indicate that the model performs consistently well across all three outcome classes—“hit”, “home_run”, and “out”—and is not disproportionately favoring the majority class. This is especially important in multiclass problems with class imbalance, as macro-averaging ensures equal treatment of all categories regardless of their frequency.

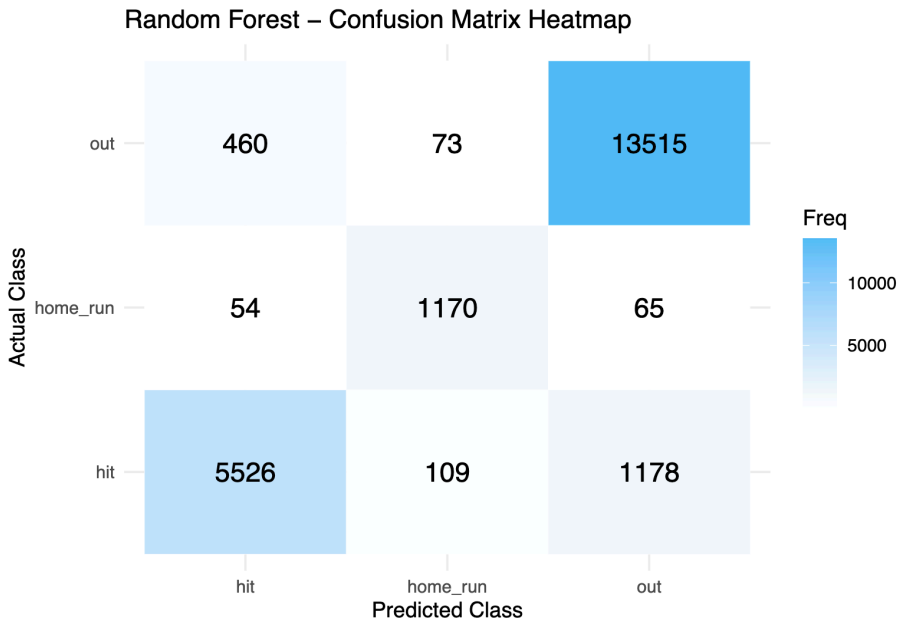

Figure 8. Confusion matrix heatmap for random forest model.

The XGBoost model also demonstrated excellent performance in predicting swing outcomes, achieving a test set accuracy of 90.8%. According to the confusion matrix (Figure 9), the model was particularly strong in predicting "out" outcomes, with a sensitivity of 96.3% and precision of 90.9%. For "home_run" outcomes, it achieved 90.2% sensitivity and 87.3% precision, suggesting reliable identification of true home runs. The "hit" category showed slightly lower sensitivity at 79.6%, but maintained high precision at 91.4%, reflecting a more cautious but accurate approach to labeling hits. Furthermore, the macro-averaged metrics support the model's balanced performance, achieving a precision of 89.9%, recall of 88.7%, and F1 score of 89.1%. These scores indicate that XGBoost maintained consistently high classification ability across all outcome classes—"hit", "home_run", and "out"—while minimizing class bias, similar to the Random Forest model.

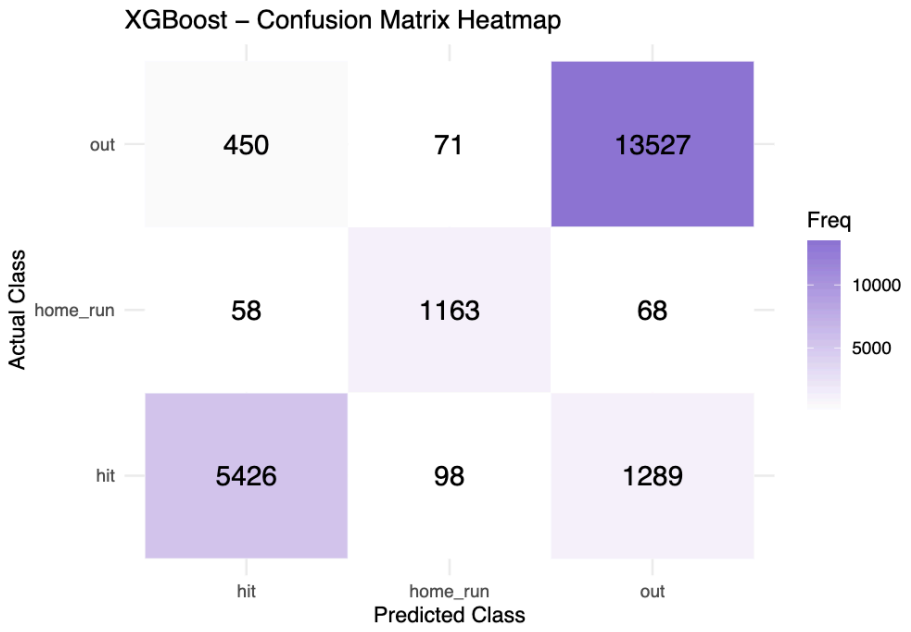

Figure 9. Confusion matrix heatmap for XGBoost model.

Finally, the Feedforward Neural Network model achieved a solid overall accuracy of 89.3% on the test set, demonstrating its effectiveness in classifying swing outcomes. As shown in the confusion matrix (Figure 10), the model performed particularly well at identifying "out" outcomes, with a sensitivity of 95.9% and a precision of 89.3%, indicating strong true positive performance on the majority class. Predictions for the "home_run" class were also strong, with a sensitivity of 89.4% and precision of 84.3%.. The "hit" class had the lowest sensitivity at 75.6%, though its precision remained high at 90.3%, suggesting the model was more cautious when labeling hits. Furthermore, the macro-averaged metrics reflect consistent performance across all outcome categories, with a precision of 88.0%, recall of 87.0%, and F1 score of 87.2%. These values demonstrate the model's ability to generalize well in a multiclass classification context, even under class imbalance. While the neural network slightly underperformed compared to Random Forest and XGBoost, its balanced classification metrics suggest it remains a competitive and interpretable model.

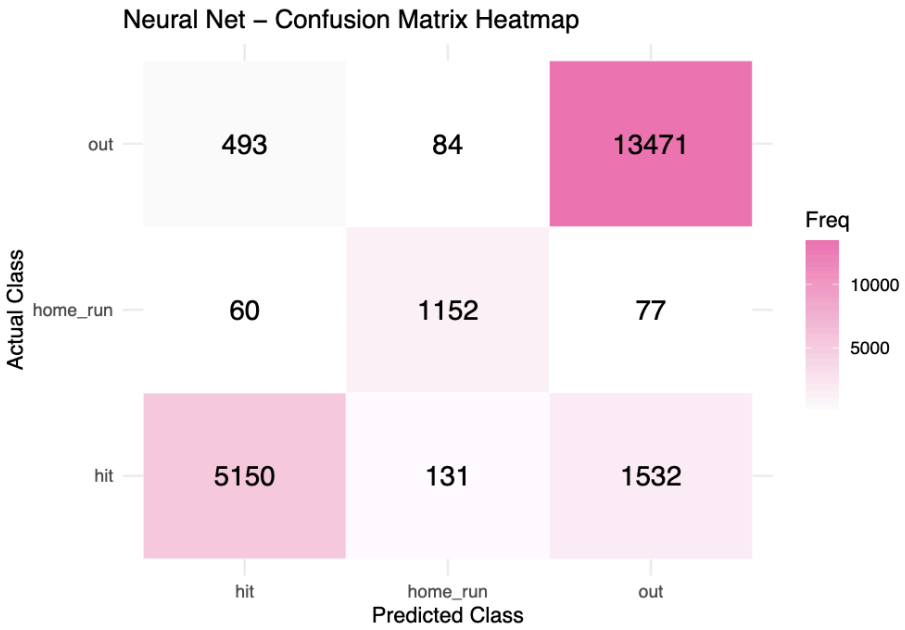

Figure 10. Confusion matrix heatmap for feedforward neural network model.

4.2. Research Question 2

Figure 11 represents the (x, y) coordinates of hit balls from a random sample (n=5000), colored by each type of hit ball. The hit coordinates are pixel coordinates of the ball using the MLB map of the field.

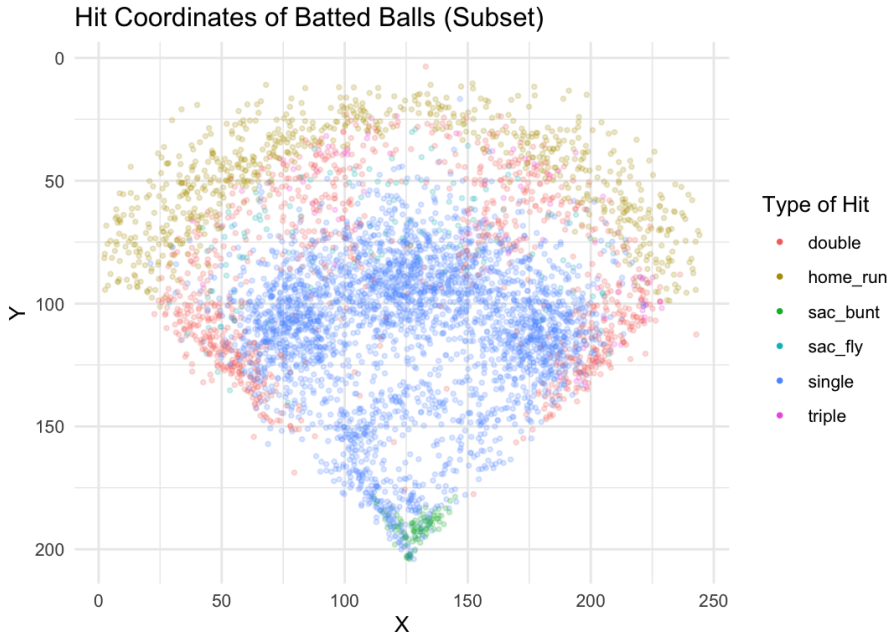

Figure 11. Hit ball coordinates colored by type of hit (n = 5000 sample).

We can see that different types of hit balls are concentrated in different parts of the field—for example, the green points representing sacrificial bunts are mostly located near the home plate, at the bottom vertex of the diamond. While there are somewhat distinct clusters of points, these clusters are non-convex, not uniform in size, and neither spherical nor elliptical, motivating our choice to implement spectral clustering.

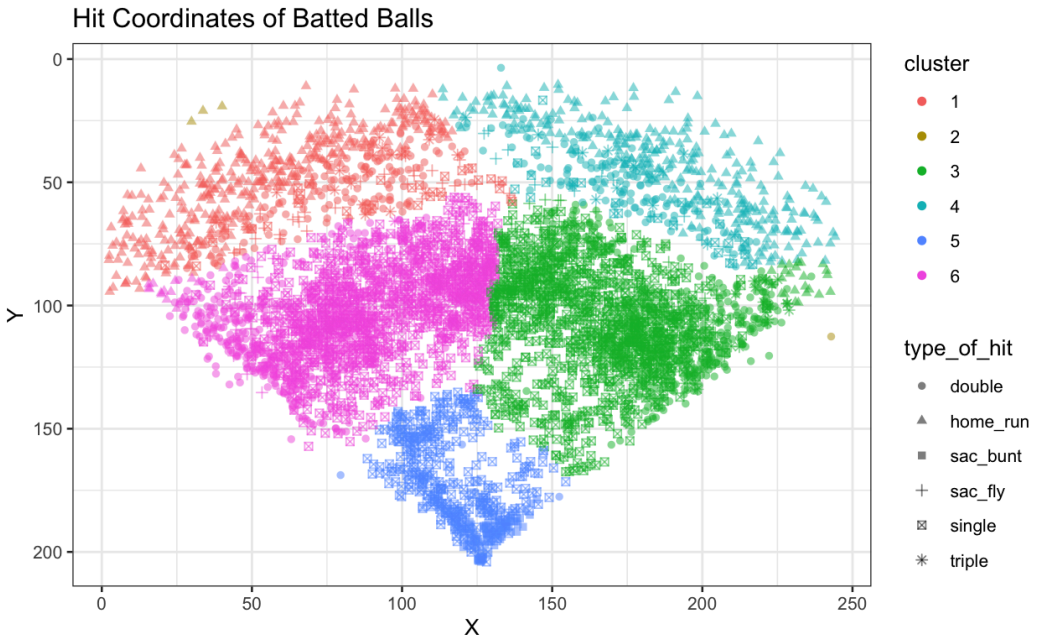

Figure 12. Spectral clustering results (K = 6) with cluster assignment and true type. The color of the points represents the cluster assignment, and the shape of the point represents the actual type of hit.

While spectral clustering did recover some of the non-convexity, especially at the top of the graph in the outfield, it did not fully recover the original clusters. It appears that doubles are perhaps the most complicated type of hit to identify, with an almost radial pattern emitting from home plate in the original data. These results may provide reason to believe that hit coordinates alone are not informative enough as it relates to hit type, and additional information (or a more advanced clustering method) may be needed.

4.3. Research Question 3

PCA requires that there be some correlation between variables to be effective. As such, before implementing our proposed pipeline (PCA → XGboost), we created a heatmap of correlation between variables, limited to numeric variables for the sake of visualization. Figure 13 demonstrates the results of this correlation analysis.

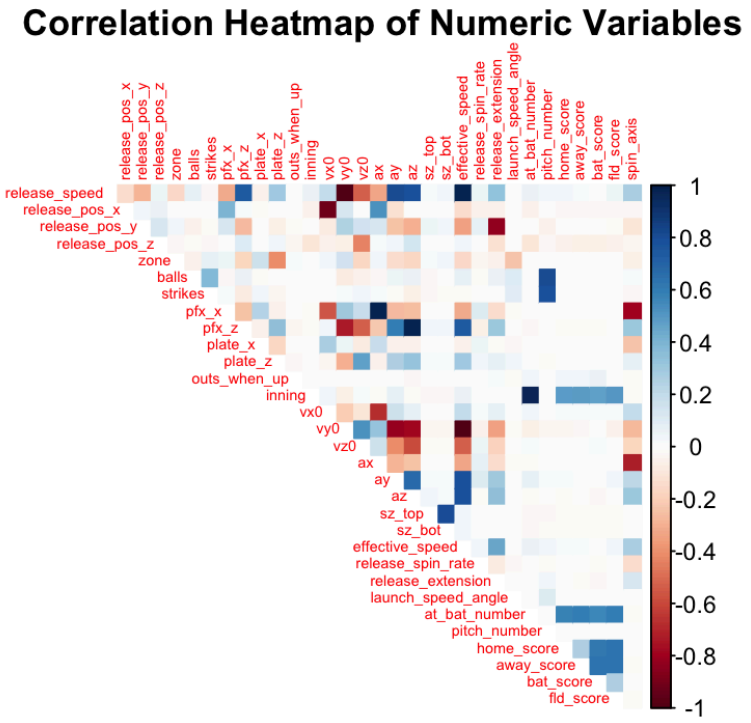

Figure 13. Correlation heatmap of numeric features for RQ3.

Darker, more saturated colors (dark blue and dark red) indicate a stronger correlation. For example, *release_speed* (pitch velocity) is highly correlated with both *pfx_z* (vertical movement of the pitch from the catcher's perspective, in feet) and *vy0* (the velocity of the pitch in the y-dimension). Overall, the correlation analysis indicates that there are linear relationships present between some of the features in the dataset.

Given that the assumptions were met, we proceeded to apply PCA. It should be noted that the numeric features and the outcome were normalized, and the categorical features were one-hot encoded. It was found that with only the first two principal components, we were able to explain 99% of the original variance in the data (Figure 14). Therefore, we proceeded with only these two components as predictors on our XGboost model.

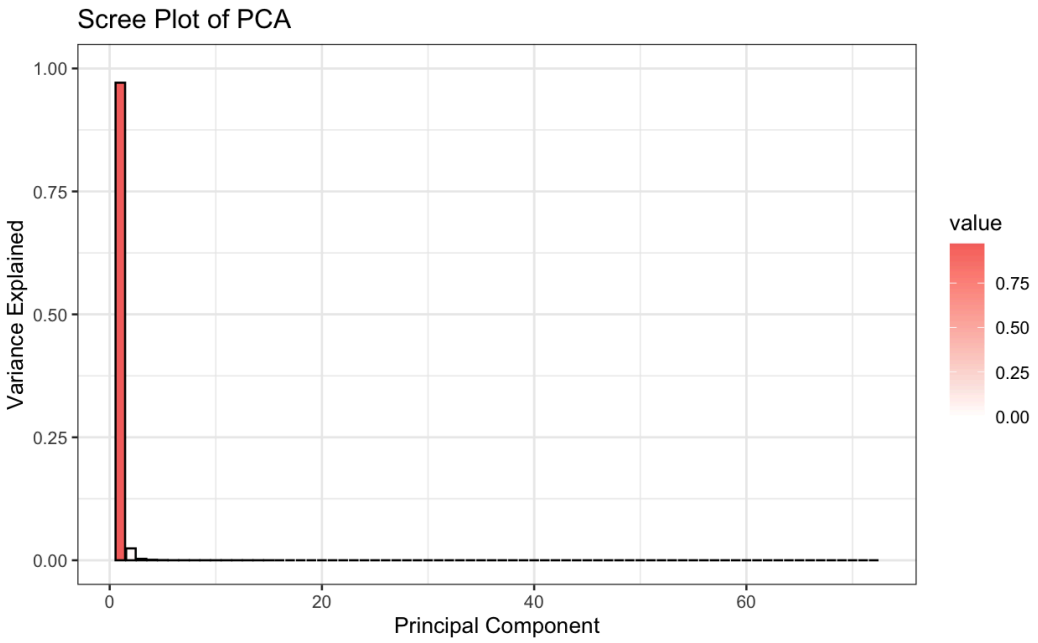

Figure 14. Scree plot of variance explained by top principal components.

The data was then split into training and testing using an 80/20 split. The hyperparameters selected were a learning rate of 0.1 to prevent overfitting—all other hyperparameters were chosen to be their default value. The model was found to have a root mean squared error measure of ~1, which given the spread of the outcome (-2.3, 6.3) demonstrates a reasonable performance.

Using the Gain measure output by XGboost, the two components are found to be of approximately equal performance in predicting wOBA. As such, the loading vectors of both components were investigated to identify key variables. The top 5 contributing variables for each component are displayed in Figure 15. Here, dot size corresponds to degree of contribution (in percent). For example, Pitch Spin Axis contributes 99% to PC2, but only 2% to PC1.

The results of this analysis indicate that pitch characteristics are by and large the most important factors contributing to weighted on-base average. Specifically, the spin axis and spin rate appear most important, with spin axis, pitch acceleration (in the x-dimension) and pitch velocity (in the y-dimension) shared by both principal components. This demonstrates that teams should focus on replicating in-game pitch scenarios to improve player outcomes like on-base batting average.

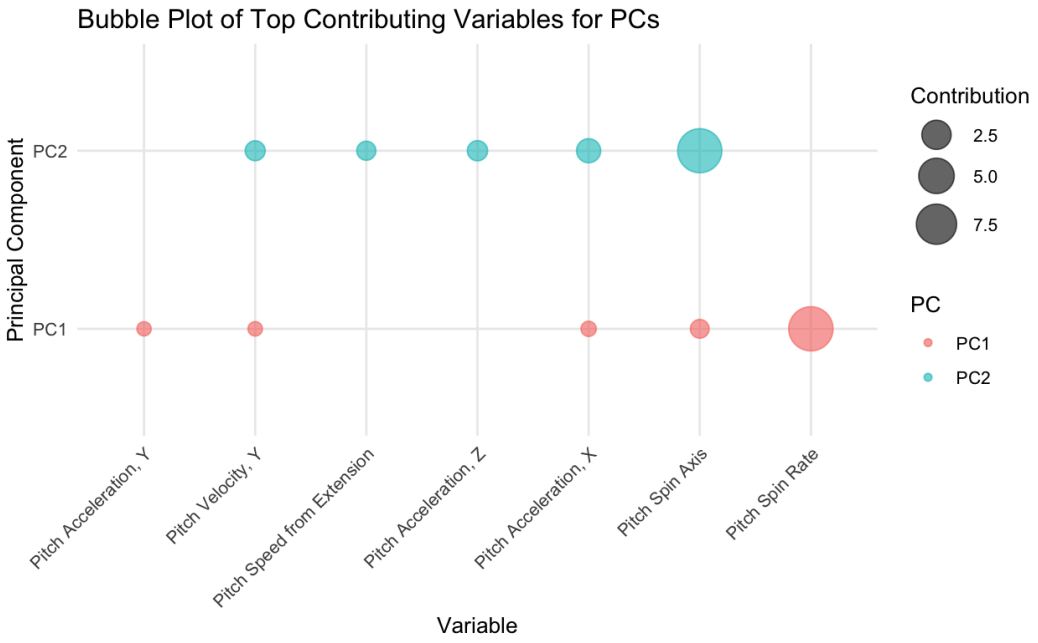

Figure 15. Bubble plot showing top contributing variables to PC1 and PC2.

5. Discussion

While Statcast data has in many ways revolutionized baseball analytics, it is not without its faults. Data from 2015–2019 relied on only camera and radar systems, whereas since the 2020 season, Hawk-Eye cameras have allowed for increased bat tracking data (swing speeds and swing paths) and other biomechanical metrics of players (MLB). However, given that our dataset was from the 2019 season, we are largely limited to features about pitch characteristics and game outcomes, leaving out data representing important parts of the game—swing characteristics, base stealing, etc.

Further, some data that is collected by Statcast has been found to be inaccurate or incomplete. The tool has been found to overestimate pitch velocity by up to 2 miles per hour, systematically miscalculate pitch coordinates in certain parks, and be unable to calculate any metrics for pop-flys or ground balls depending on where the detector was placed (Arthur, 2017; Kagan, 2006). Some of these limitations directly apply to our analysis—specifically, baseball fans are skeptical of the hit coordinates of balls for certain outcomes. For example, in the 2019 data, there are doubles recorded beyond the home-run range (potentially beyond the field fence), which would be extremely unlikely in a game

setting. Since we cannot be sure if hit coordinates are incorrect or outcomes are incorrectly labeled, we cannot conclusively evaluate the efficacy of our clustering algorithm.

One limitation of the analysis for RQ1 is the manual selection of swing features based on domain intuition rather than a formal feature selection process. While this approach helps focus the models on well-known predictors (such as launch speed, launch angle, and plate location), it may exclude other variables that hold predictive value or interact with selected features in complex ways. A more comprehensive strategy would involve starting with a broader set of available features and applying automated selection techniques—such as LASSO regularization or Elastic Net—to objectively identify the most informative predictors. This would enhance reproducibility, reduce potential bias, and allow the models to fully leverage the richness of the Statcast dataset.

Another challenge relates to RQ2, where we attempted to identify hit type using spectral clustering on coordinate data. The clustering algorithm did not fully recover the original labels, which may be due in part to label noise or measurement error in the hit coordinates. Spectral clustering is also sensitive to hyperparameter choices (e.g., similarity kernel and neighborhood size), which were selected heuristically. Future work could explore more robust unsupervised methods—such as density-based clustering or deep learning–based clustering approaches—and evaluate performance using established cluster validity indices.

For RQ3, one limitation is the interpretability of principal components. While PCA effectively reduced dimensionality and identified a small number of dominant components, the transformation makes it difficult to directly map back to the original predictors. Although we visualized loading contributions to interpret key variables, some principal components may still represent complex combinations that are not easily actionable. Future extensions could consider hybrid techniques that retain interpretability, such as sparse PCA or supervised feature importance frameworks.

Finally, across all research questions, models were trained and evaluated on a single season of data. This decision was made to ensure consistency in data collection technology and reduce computational overhead. However, limiting the analysis to one season restricts the generalizability of the results, particularly if league dynamics, player performance trends, or data recording systems

evolved over time. Future studies could extend this work by incorporating data from multiple seasons and evaluating model stability across time. Temporal cross-validation or nested modeling frameworks may also help assess how well predictive relationships hold in different years or across varying league conditions.

References

Albert, J. (2018). *Exploring baseball data with R*. Chapman and Hall/CRC.

Arthur, R. (2016). Who's hitting the ball harder... and who's just getting lucky? *FiveThirtyEight*. Retrieved from https://fivethirtyeight.com/features/baseballs-best-mlb-hitters-statcast-data/

Arthur, R. (2017). Baseball's new pitch-tracking system is just a bit outside. *FiveThirtyEight*. Retrieved from https://fivethirtyeight.com/features/baseballs-new-pitch-tracking-system-is-just-a-bit-outside/

Brandon, W. (n.d.). Decision tree, random forest, and XGBoost: An exploration into the heart of machine learning. *Medium*. Retrieved from https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-90dc212f4948

Daley-Harris, M. (2016). Fixing batted-ball statistics with Statcast. *The Hardball Times*. Retrieved from https://tht.fangraphs.com/fixing-batted-ball-statistics-with-statcast/

DeepAI. (n.d.). Feed-forward neural network. Retrieved from https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network

Hammack, B. (n.d.). *Predict at-bat outcome* [GitHub repository]. Retrieved from https://github.com/bjhammack/predict-at-bat-outcome/

Kagan, D. (2006). The physics of Statcast errors. *The Hardball Times*. Retrieved from https://tht.fangraphs.com/the-physics-of-statcast-errors/

Kaggle. (2022). *MLB Statcast Data* [Dataset]. Retrieved from https://www.kaggle.com/datasets/s903124/mlb-statcast-data

Mann, E. (2024). Clustering hitters: Using k-means for MLB hitters. *Medium*. Retrieved from https://elijahmann.medium.com/mlb-hitters-kmeans

MLB. (n.d.). *Statcast glossary*. Retrieved from https://www.mlb.com/glossary/statcast

Perpetua, A. (2017). xStats and fantasy uses for Statcast. *The Hardball Times*. Retrieved from https://tht.fangraphs.com/xstats-and-fantasy-uses-for-statcast/

Petti, B. (2016). Using Statcast data to predict hits. *The Hardball Times*. Retrieved from https://tht.fangraphs.com/using-statcast-data-to-predict-hits/

Reddit. (2018). Interpreting hc_x and hc_y (Statcast). *r/Sabermetrics*. Retrieved from https://www.reddit.com/r/Sabermetrics/comments/8efn3y/interpreting_hc_x_and_hc_y_statcast/

Sharpe, S. (2017). An introduction to xwOBA. *MLB Technology Blog*. Retrieved from http://m.mlb.com/news/article/275882348/statcast-introduces-xwoba/

Spiceworks. (n.d.). XGBoost vs. random forest vs. gradient boosting: Key differences explained. Retrieved from https://www.spiceworks.com/tech/artificial-intelligence/articles/xgboost-vs-random-forest-vs-gradient-boosting/