**Machine Learning: Umpire Analytics**


By: Rohan Singh, Angela Bauer, Zachary Foster

MEEN 423-500

Fall 2023

**Table of Contents**

**Introduction**

As more pitching technology is introduced to the game of baseball, propositions have been made to remove human umpires and replace them with computers. However, some fans and players still argue that implementing automated strike zones would remove the "human element" of the game. This report covers an analysis of pitch strike zone data that aims to create a machine learning model replicating the "human element" of baseball umpires.

**Objective**

Unlike current pitching technology that aims to determine the correctness of a call, the objective of this analysis is to determine and replicate individual strike zones of six Major League Baseball (MLB) umpires: Angel Hernandez, Erich Bacchus, Junior Valentine, Malachi Moore, Pat Hoberg, Quinn Wolcott.

**Data Processing**

The pitch strike zone data utilized in this analysis was collected from Baseball Savant and 35508 pitches from 170 games in the 2023 regular MBL season. **Table 1** shows the number of data points collected by umpire.

**Table 1:** Number of pitch strike zone data points collected from Baseball Savant by umpire.

| Umpire | Games | Pitches |
|---|---|---|
| Angel Hernandez | 13 | 2769 |
| Erich Bacchus | 32 | 6561 |
| Junior Valentine | 35 | 7092 |
| Malachi Moore | 31 | 6603 |
| Pat Hoberg | 29 | 6417 |
| Quinn Wolcott | 30 | 6066 |
| **Total** | 170 | 35508 |

The gameplay data used in this analysis only included pitches where the batter did not swing. **Table 2** explains all seven variables used to define each pitch.

**Table 2:** Pitch strike zone data variable explanations.

| Variable | Explanation |
|---|---|
| $gameID$ | Identification number of the game |
| $pCount$ | Order the pitch was delivered in the game |
| $pX$ | X-position of the pitch |
| $pZ$ | Z-position of the pitch |
| $pCall$ | Umpire's call (Strike or Ball) |
| $sZ_{top}$ | Z-position of the top of the batter's strike zone |
| $sZ_{bot}$ | Z-position of the bottom of the batter's strike zone |
| $gameHP$ | Umpire's name |

The $pX$ and $pZ$ of the pitch are recorded when the baseball passes home plate. The point of origin for all measurements is the middle of home plate on the ground, as seen in **Figure 1**. The positive X-direction is

to the umpire's right and pitches above the ground are positive. If a pitch bounces before reaching home plate, the Z-measurement is negative.

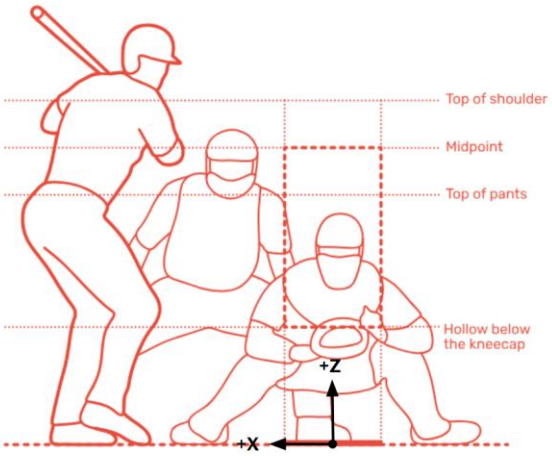


**Figure 1:** Visual representation of the strike zone and origin.

The horizontal strike zone is defined by the edges of home plate, plus the radius of a baseball on either side. The width of home plate is 17 inches, and the radius of a baseball is 0.12 feet. Therefore, the strike zone boundary for every batter is +0.83 ft and -0.83 ft in the X-direction. The vertical strike zone is defined by $sZ_{top}$ and $sZ_{bot}$. A batter's $sZ_{top}$ is defined by the midpoint between the top of their shoulder and the of top their pants. A batter's $sZ_{bot}$ is defined by the hollow below their kneecap. Both measurements are determined before the game and constant, regardless of player movement.

Since every player has a unique vertical strike zone, the Z-position in for each pitch was normalized based on the average $sZ_{top}$ and $sZ_{bot}$ in the MBL. The normalized Z-position was calculated using **Equation 1**.

$$\frac{pZ - sZ_{bot}}{sZ_{top} - sZ_{bot}}\left(sZ_{top,avg} - sZ_{bot,avg}\right) + sZ_{bot,avg} = pZ_{norm} \qquad \textbf{Equation 1}$$

where $sZ_{top,avg}$ is the average $sZ_{top}$ and $sZ_{bot,avg}$ is the average $sZ_{bot}$. The $pX$ and $pZ_{norm}$ were used to plot all of the pitches along with a rectangle representing the strike zone. The pitches called by Angel Hernandez are shown in **Figure 2**.

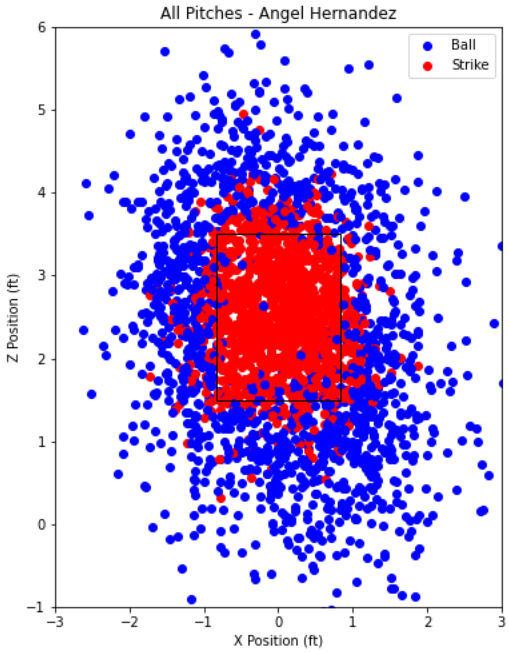


**Figure 2:** Visual representation of the pitches called by Angel Hernandez.

The plots showing the pitches for the remaining umpires are included in **Appendix A**. The pitch data for each umpire was then split into 95% training and 5% testing subsets. These subsets were used to train and test three different modes.

## Machine Learning Methods

### *Support Vector Machine*

A Support Vector Machine (SVM) is a supervised machine learning algorithm that aims to find a hyperplane that separates data into binary classes while attempting to maximize the margin between them. The margin is the distance between the hyperplane and the nearest data points from each class. SVM is effective for both linearly and non-linearly separable data; however, this application requires a non-linear radial based approach. The following line was used to implement the SVM model: SVC(kernel='rbf', C=1.0, gamma='scale'). Parameter 'C' refers to the tradeoff between correct classification and the maximization of the decision margin. Parameter 'gamma' refers to how far a single training example reaches. The SVM model and test pitches called by Angel Hernandez are shown in **Figure 3**. The plots showing the SVM models for the remaining umpires are included in **Appendix B**.

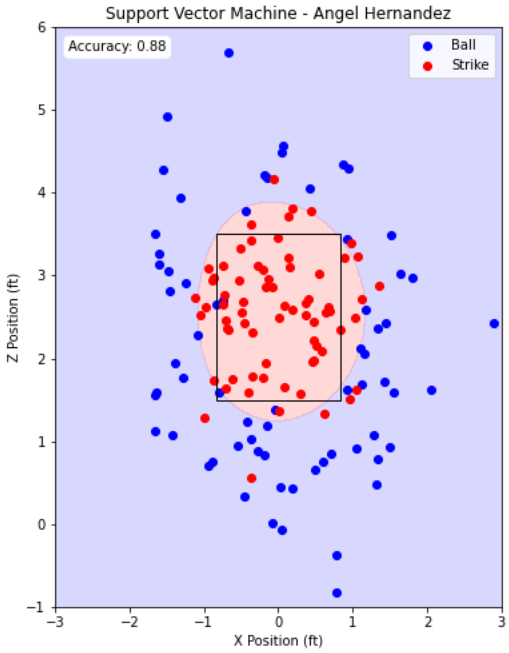


**Figure 3:** Support Vector Machine model for Angel Hernandez with test pitches.

***Gradient Boosting Classifier***

A Gradient Boosting Classifier (GBC) an ensemble learning algorithm that combines the predictions of multiple weak learners, often decision trees, to create a strong predictive model. Every additional weak learner focuses on minimizing the residuals of the ensemble up to that point. Gradient Boosting aims to improve accuracy by sequentially refining predictions and adjusting for error in the model. The final model is a sum of the weak learners, weighted by their assigned learning rate. The following line was used to implement the SVM model: GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=None). Parameter n_estimators is the number of boosting stages performed. The learning_rate parameter scales the contribution of each weak learner and max_depth limits the number of nodes in the tree. The GBC model and test pitches called by Angel Hernandez are shown in **Figure 4**. The plots showing the GBC models for the remaining umpires are included in **Appendix C**.

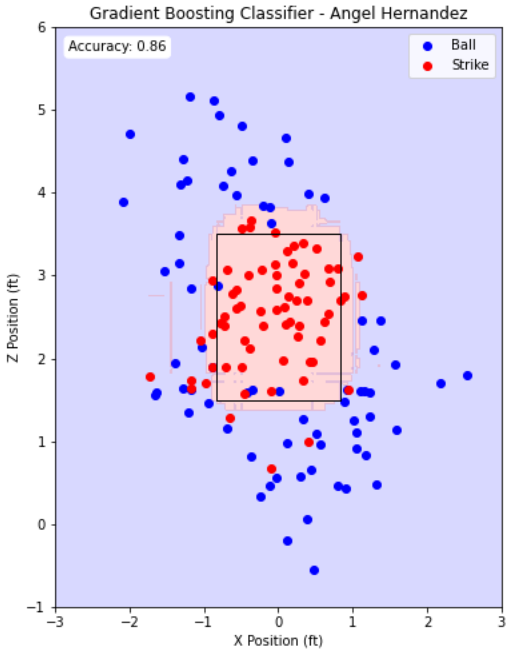


**Figure 4:** Gradient Boosting Classifier model for Angel Hernandez with test pitches.

***Random Forest Classifier***

A Random Forest Classifier (RFC) is an ensemble learning method that builds multiple decision trees during training. Each tree is constructed on a subset of the dataset, and at each split, a random subset of features is considered. The RFC minimizes overfitting and increases accuracy by implementing a majority voting system. The following line was used to implement the RFC model: RandomForestClassifier(n_estimators=200, max_depth=10, min_samples_split=10, min_samples_leaf=1). Parameter n_estimators defines the number of trees in the forest and max_depth is the maximum depth of the tree. The min_samples_split parameter is the minimum number of samples required to split an internal node. The min_samples_leaf is the minimum number of samples required to be at a leaf node. The RFC model and test pitches called by Angel Hernandez are shown in **Figure 5**. The plots showing the RFC models for the remaining umpires are included in **Appendix D**.

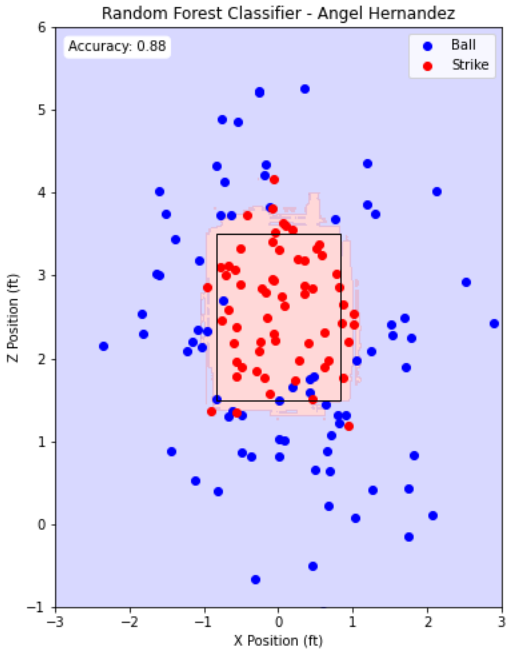


**Figure 4:** Random Forest Classifier model for Angel Hernandez with test pitches.

**Decision Boundary Visualization**

The contour plots shown in **Figures 2-4** were created to visualize and compare the decision boundary of all three model types. The test pitches were overlaid to easily identify discrepancies between the actual and predicted calls. The accuracies of the models' ability to replicate the umpires' calls were included in the plots.

**Optimization**

***Hyperparameter Tuning***

Hyperparameter tuning is the process of finding the optimal set of parameters for a machine learning model. The GridSearchCV process was applied to the Random Forest Classifier to search the grid of parameters shown in **Table 3** below.

**Table 3:** Parameter grid used in the hyperparameter tuning process for the Random Forest Classifier.

| Parameter | Grid |
|---|---|
| n_estimators | [50, 100, 200] |
| max_depth | [5, 10, 20] |
| min_samples_split | [2, 5, 10] |
| min_samples_leaf | [1, 2, 5] |

The result of the hyperparameter tuning process is shown in **Table 4**. These results represent the parameters used to create a more optimized model and achieve better performance.

**Table 4:** Results of the hyperparameter grid tuning process for the Random Forest Classifier.

| Umpire | max_depth | min_samples_leaf | min_samples_split | n_estimators |
|---|---|---|---|---|
| Angel Hernandez | 10 | 1 | 10 | 200 |
| Erich Bacchus | 10 | 1 | 5 | 50 |
| Junior Valentine | 20 | 5 | 10 | 100 |
| Malachi Moore | 10 | 5 | 2 | 100 |
| Pat Hoberg | 10 | 5 | 5 | 100 |
| Quinn Wolcott | 5 | 1 | 10 | 50 |

**Results**

The accuracy of the SVM, GBC, and RFC models for every umpire are shown in **Table 5**.

**Table 5:** Average accuracy for all machine learning methods tested.

| | Support Vector Machine | Gradient Boosting Classifier | Random Forest Classifier |
|---|---|---|---|
| Angel Hernandez | 0.88 | 0.86 | 0.88 |
| Erich Bacchus | 0.86 | 0.88 | 0.88 |
| Junior Valentine | 0.90 | 0.87 | 0.87 |
| Malachi Moore | 0.88 | 0.88 | 0.93 |
| Pat Hoberg | 0.91 | 0.91 | 0.88 |
| Quinn Wolcott | 0.87 | 0.90 | 0.90 |
| **Average** | **0.884** | **0.883** | **0.890** |

The accuracy values show that all machine learning methods performed approximately the same. The results indicate that any of the three methods tested can predict the unique strike zone of an umpire with a accuracy between 0.883 and 0.890.

**Future Applications**

The application of machine learning models in baseball umpiring introduces intriguing possibilities. Firstly, these models would be used to test how alternative umpires might have handled contentious or close calls. These models could also be used to simulate complete games under different strike zones. This could help predict the outcome of the game if called by a different umpire. Finally, the models can contribute to understanding umpire behavior. For example, the models could be used to evaluate whether an umpire widened their personal strike zone intentionally to influence the conclusion of a game. The use of machine learning in these scenarios opens avenues for insightful analysis and enhances our understanding of the dynamics within baseball officiating.

**Conclusions**

This report presented an analysis of pitch strike zone data with the objective of creating a machine learning model to replicate the "human element" of baseball umpires. The study focused on determining and replicating individual strike zones of six Major League Baseball (MLB) umpires. The data processing involved normalizing Z-positions based on the average strike zone measurements in the MLB. The three machine learning models—Support Vector Machine (SVM), Gradient Boosting Classifier (GBC), and

Random Forest Classifier (RFC)—were implemented and tested for accuracy. The results indicated that all three methods performed similarly, achieving accuracies between 0.883 and 0.890. The report discussed the potential future applications of these models in testing alternative umpire decisions, simulating complete games, and understanding umpire behavior.

## Appendix

### *Appendix A: Pitch Visualization*

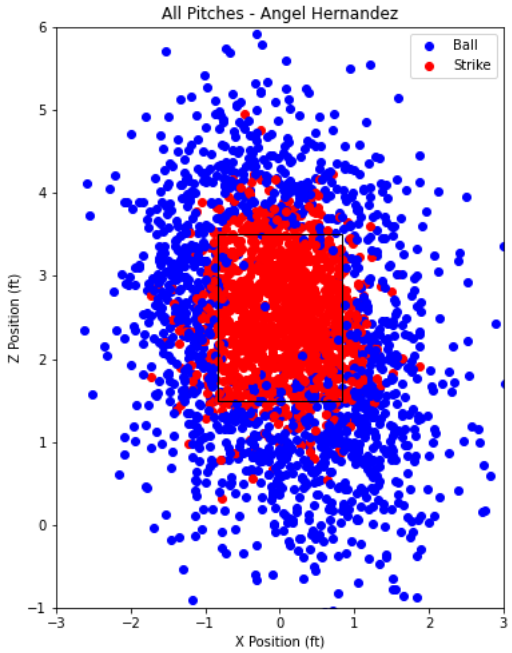


**Figure A.1:** Pitches called by Erich Bacchus.

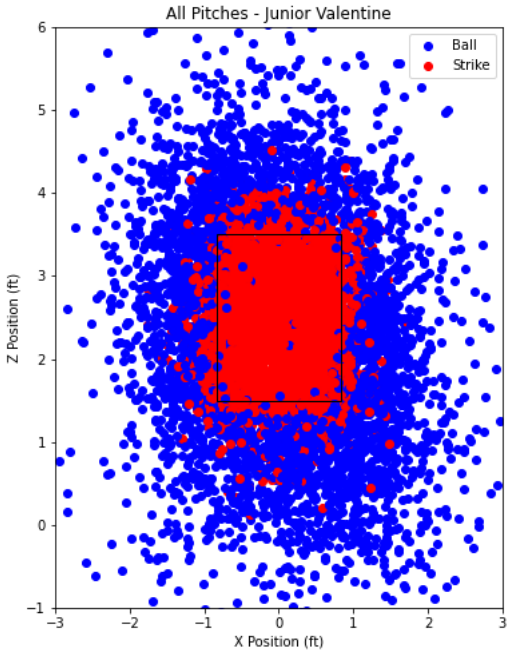


**Figure A.2:** Pitches called by Junior Valentine.

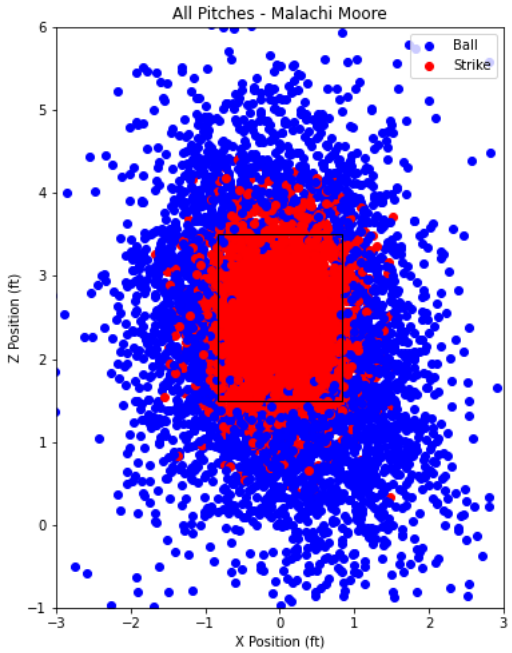


**Figure A.3:** Pitches called by Malachi Moore.

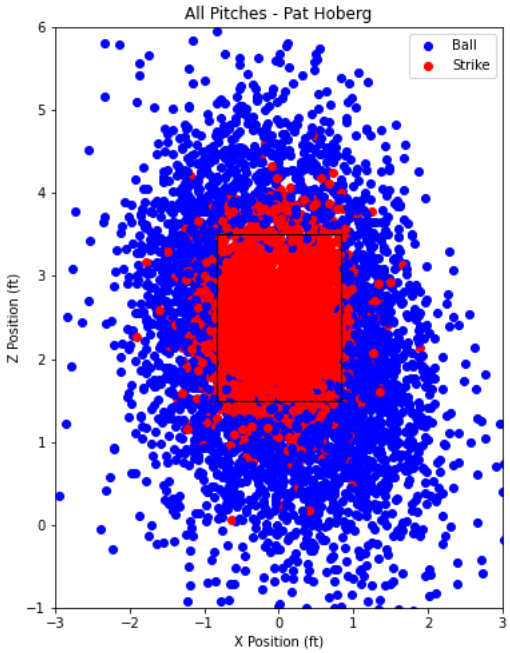


**Figure A.4:** Pitches called by Pat Hoberg.

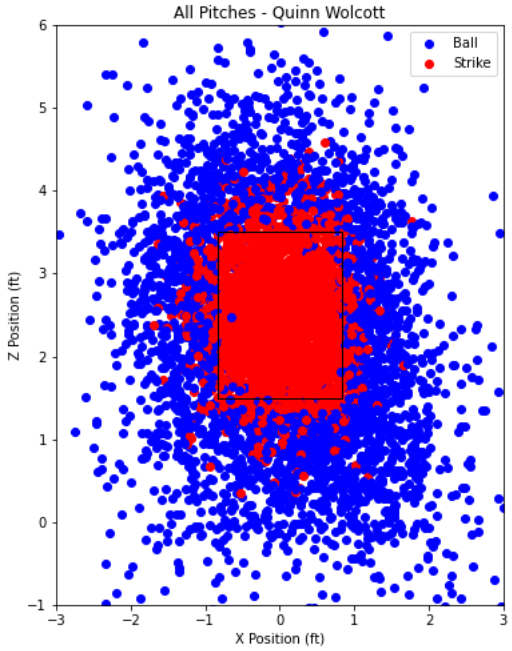


**Figure A.5:** Pitches called by Quinn Wolcott.

## *Appendix B: Support Vector Machine Visualization*

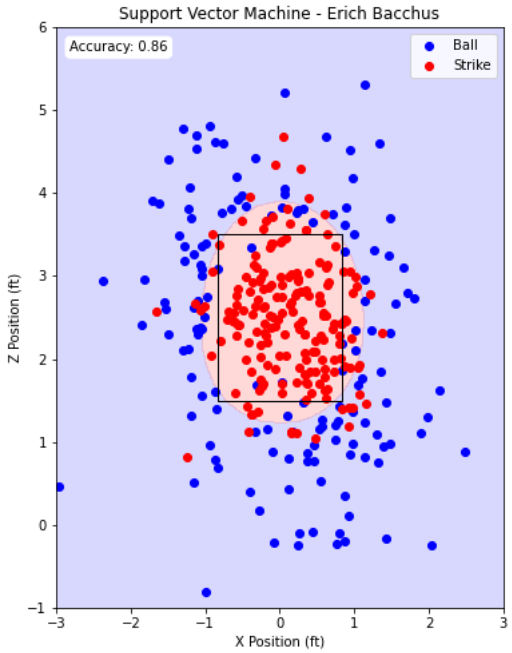


**Figure B.1:** SVM for Erich Bacchus.

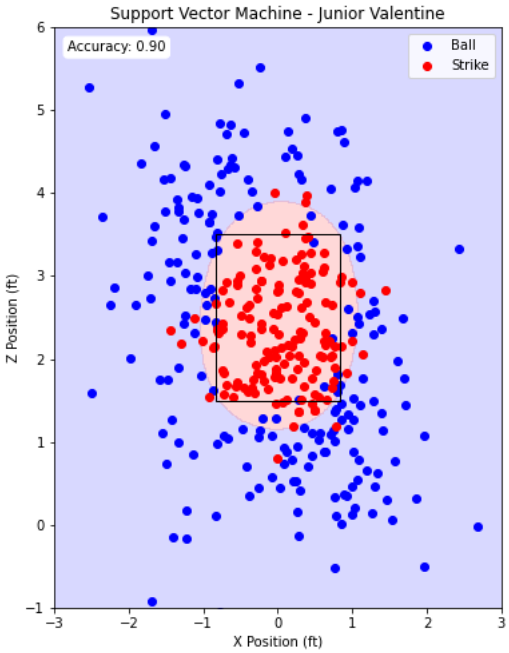


**Figure B.2:** SVM for Junior Valentine.

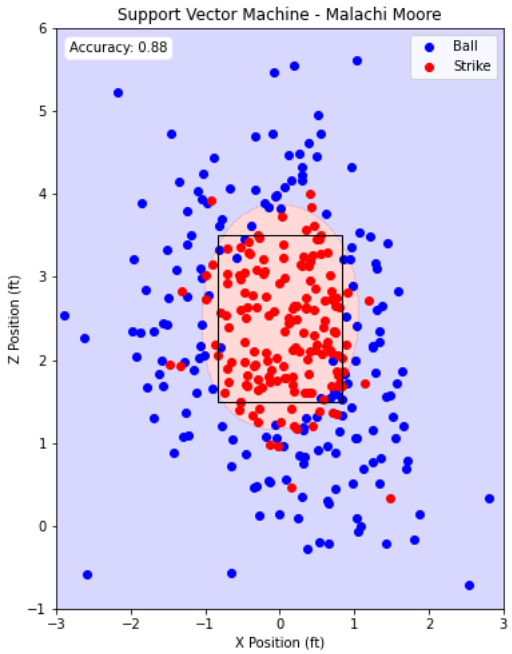


**Figure B.3:** SVM for Malachi Moore.

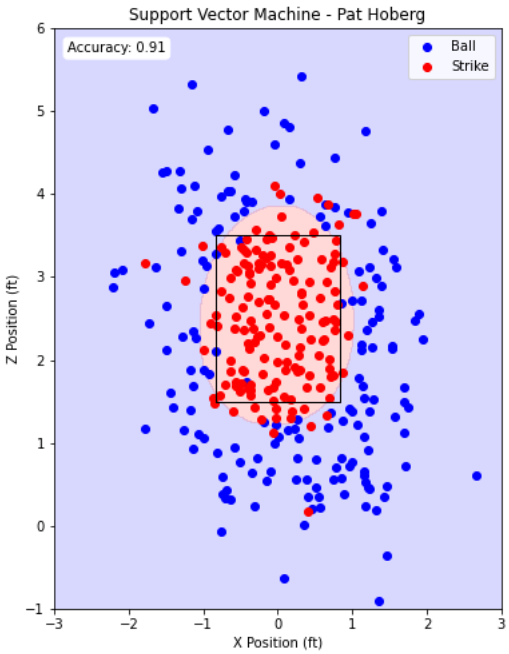


**Figure B.4:** SVM for Pat Hoberg.

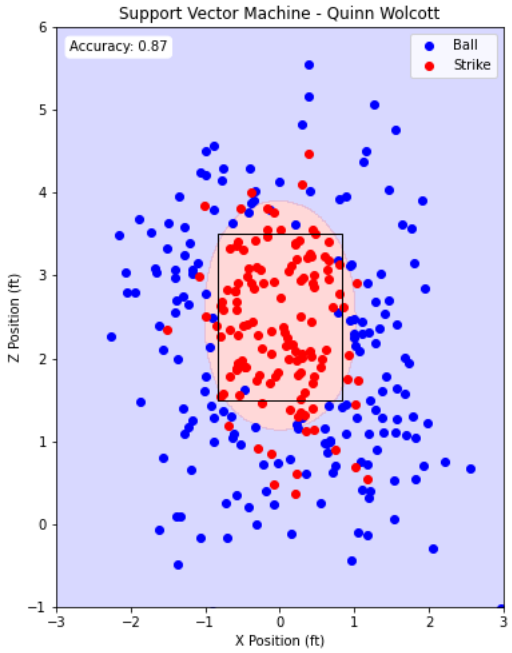


**Figure B.5:** SVM for Quinn Wolcott.

## *Appendix C: Gradient Boosting Classifier Visualization*

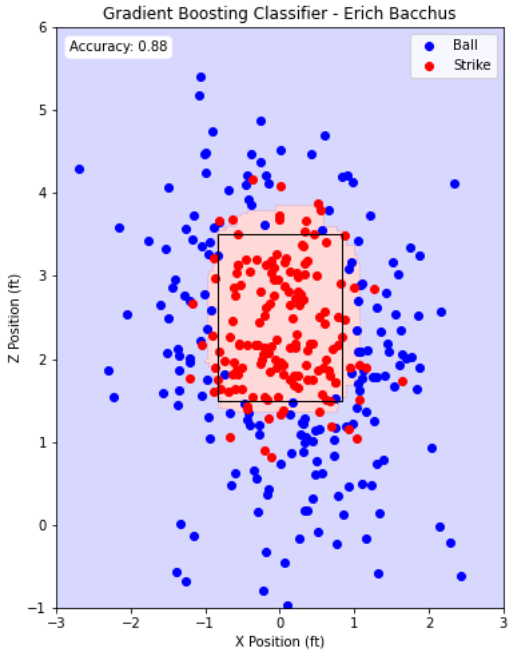


**Figure C.1:** GBC for Erich Bacchus.

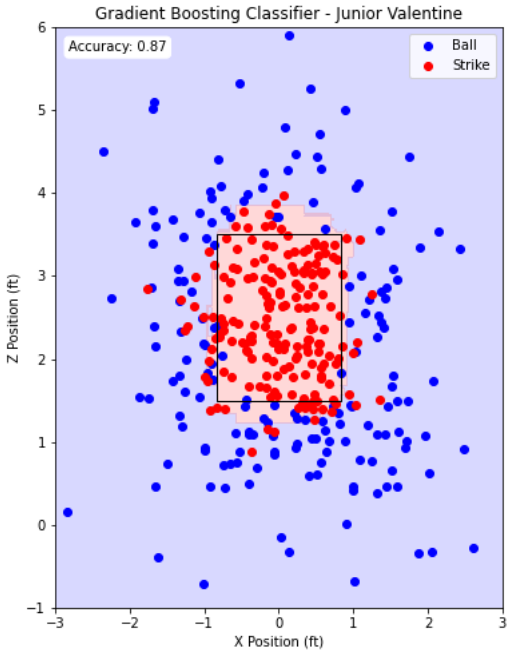


**Figure C.2:** GBC for Junior Valentine.

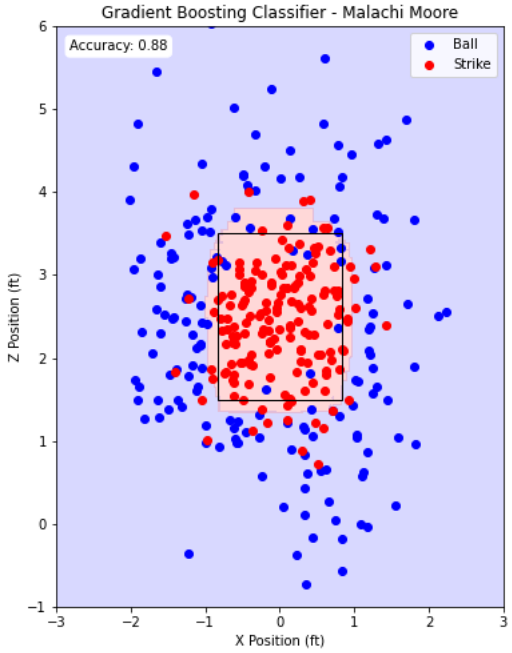


**Figure C.3:** GBC for Malachi Moore.

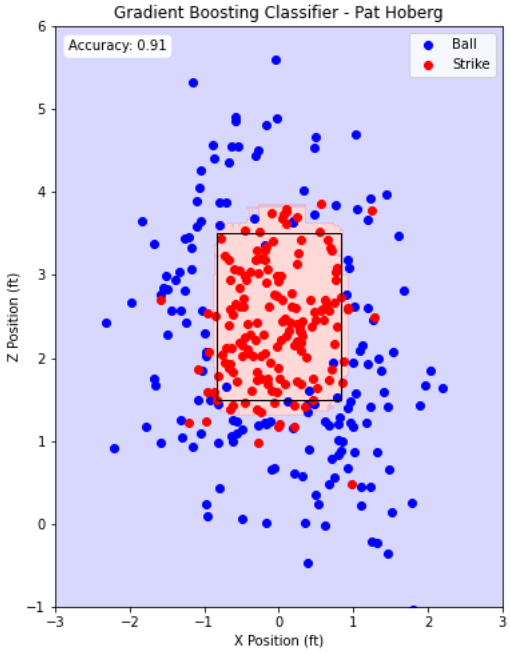


**Figure C.4:** GBC for Pat Hoberg.

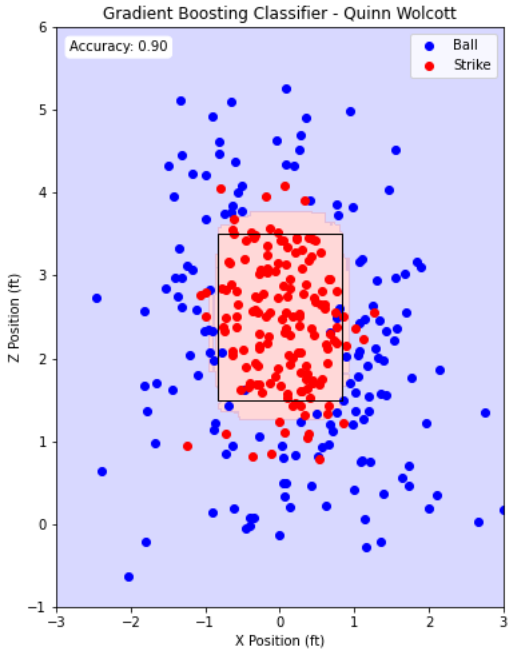


**Figure C.5:** GBC for Quinn Wolcott.

## *Appendix D: Random Forest Classifier Visualization*

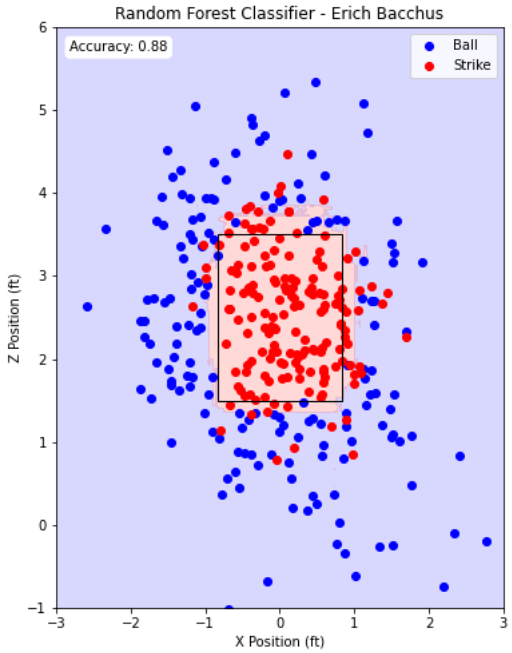


**Figure D.1:** RFC for Erich Bacchus.

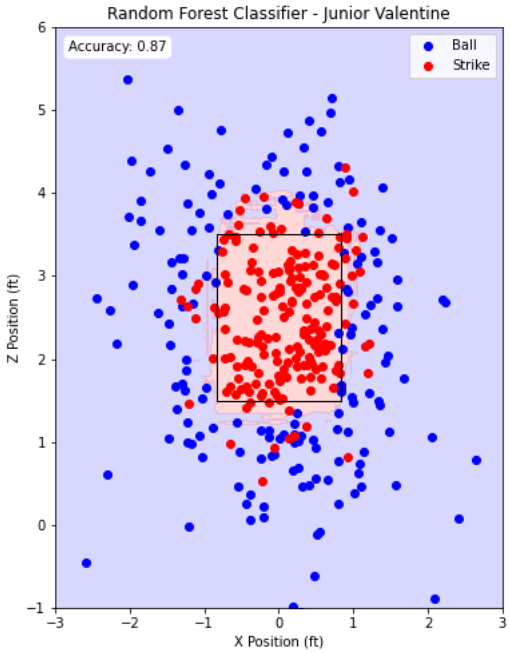


**Figure D.2:** RFC for Junior Valentine.

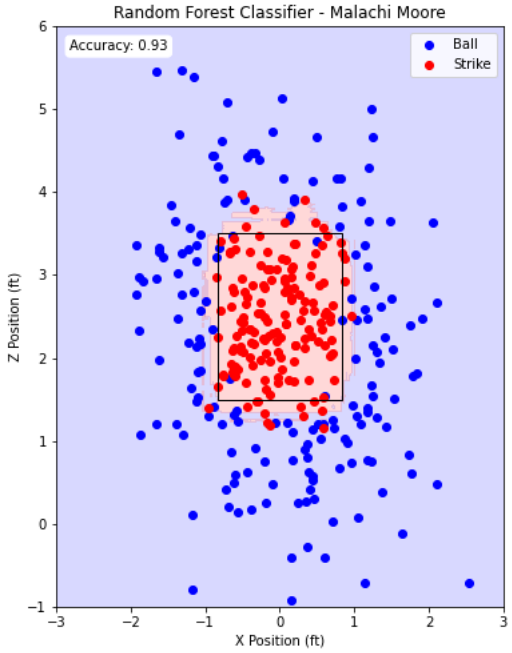


**Figure D.3:** RFC for Malachi Moore.

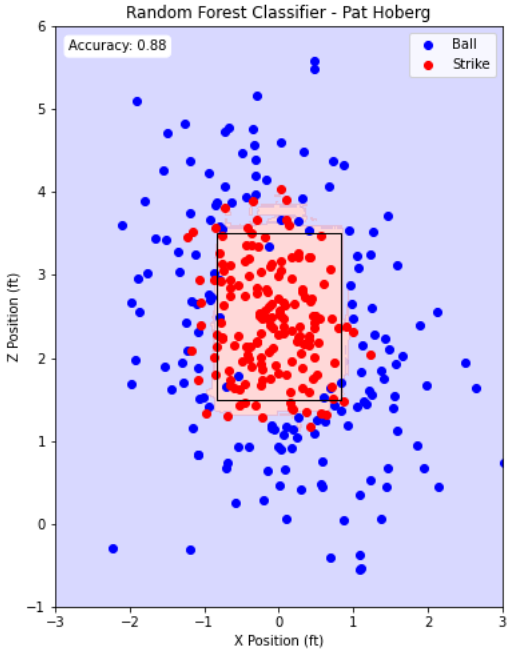


**Figure D.4:** RFC for Pat Hoberg.

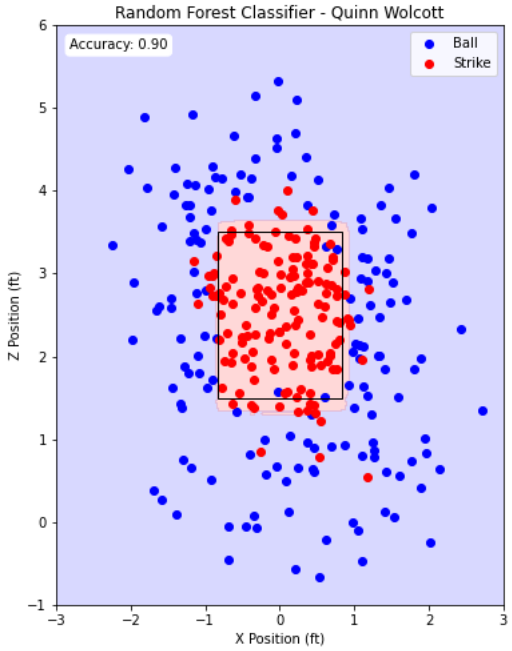


**Figure D.5:** RFC for Quinn Wolcott.

*Appendix E: Hyperparameter Tuning Results*

**Best Hyperparameters for Angel Hernandez:** {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}

**Best Hyperparameters for Erich Bacchus:** {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}

**Best Hyperparameters for Junior Valentine:** {'max_depth': 20, 'min_samples_leaf': 5, 'min_samples_split': 10, 'n_estimators': 100}

**Best Hyperparameters for Malachi Moore:** {'max_depth': 10, 'min_samples_leaf': 5, 'min_samples_split': 2, 'n_estimators': 100}

**Best Hyperparameters for Pat Hoberg: {**'max_depth': 10, 'min_samples_leaf': 5, 'min_samples_split': 5, 'n_estimators': 100}

**Best Hyperparameters for Quinn Wolcott:** {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}