MECHANICAL ENGINEERING
TEXAS A&M UNIVERSITY

Project Presentation: Umpire Analytics

Angela Bauer | Rohan Singh | Zachary Foster

MEEN 423 - 500

December 4th, 2023

Agenda

1. Introduction
2. Objective
3. Data Loading and Preprocessing
4. Data Visualization and Separation
5. Machine Learning Methods
6. Decision Boundary Visualization
7. Optimizations
8. Results
9. Future Applications

Introduction

- **More pitching technology is being introduced to the game of baseball**
  - Propositions have been made to replace human umpires with computers
  - Some fans and players argue that automated strike zones would remove the “human element” of the game
    - "If I had a choice, I would definitely say keep the human element of the game. I just enjoy that and if a guy is missing inside the plate you can use it to your advantage or maybe that's the reason why you lost.” - Mookie Betts (LA Dodgers)

Objective

- Determine and replicate the individual strike zones of six Major League Baseball umpires
  - Accuracy is related to the model's ability to replicate the "human element" and tendencies of a particular umpire

- Accuracy is NOT related to the model's ability to categorize balls and strikes correctly

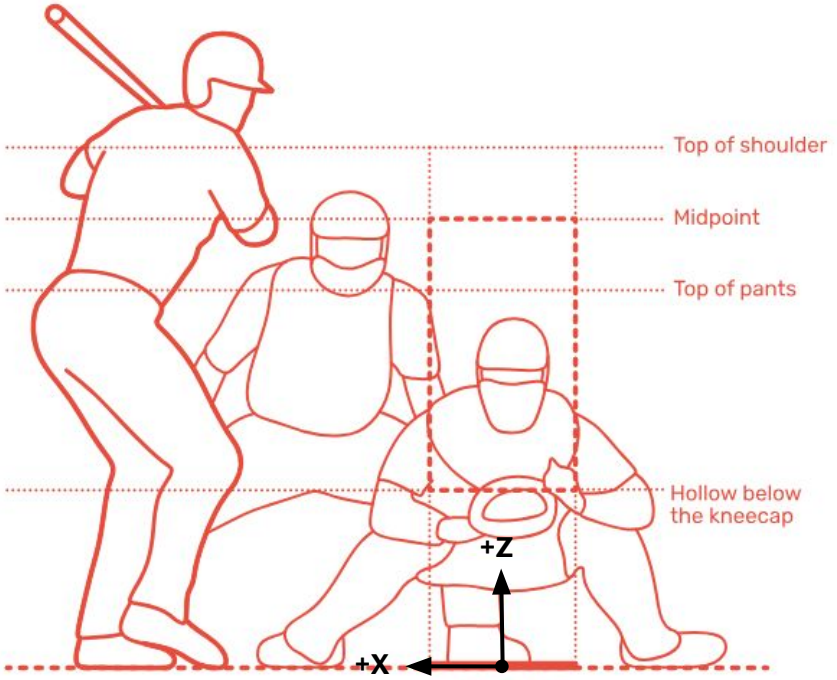
Data Loading and Preprocessing

- Baseball Savant
  - 6 Umpires
    - Every game at HP during the 2023 regular season
  - Pitch Strike Zone Data
    - Only used pitches where the batter did not swing

	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

Data Loading and Preprocessing

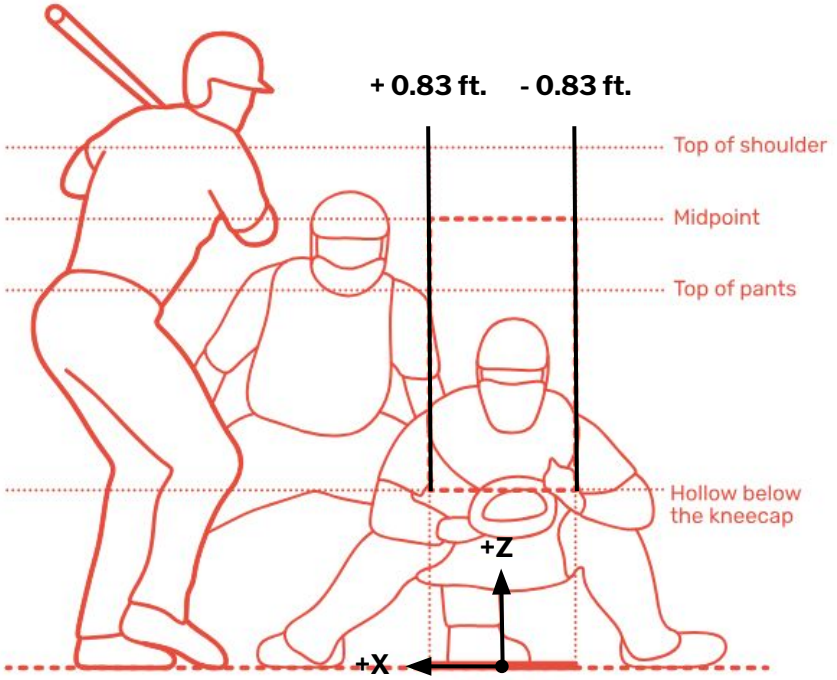
- X Position
  - Origin is at the center of HP
  - Pitches to the umpire's right are positive
- Z Position
  - Origin is at the ground
  - Pitches above the ground are positive
  - Pitches that bounce before reaching HP are negative



Data Loading and Preprocessing

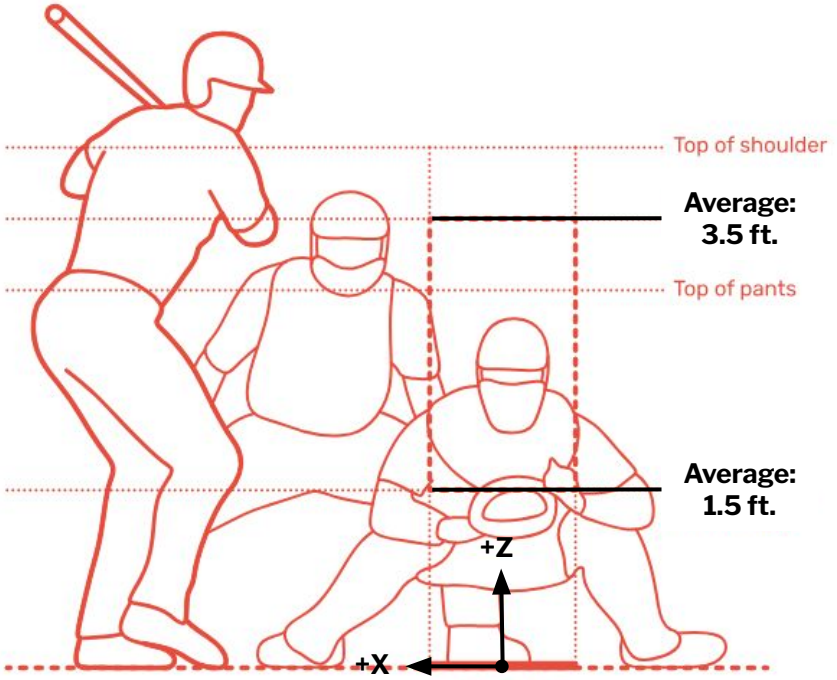
- X Position Variables
  - sX_right = + 0.83 ft.
  - sX_left = - 0.83 ft.
  - pX = X location of pitch

$$\underbrace{\left(\frac{17\ in.}{2}\right)\left(\frac{1\ ft.}{12\ in.}\right)}_{\text{Half the width of home plate}} + \underbrace{0.12\ ft.}_{\text{Radius of baseball}} = 0.828\ ft. \approx 0.83\ ft.$$



Data Loading and Preprocessing

- Z Position Variables
  - Vertical strike zone is unique to each batter
  - MLB Averages
    - sZ_top = 3.5 ft.
    - sZ_bot = 1.5 ft.
  - pZ = Z location of pitch

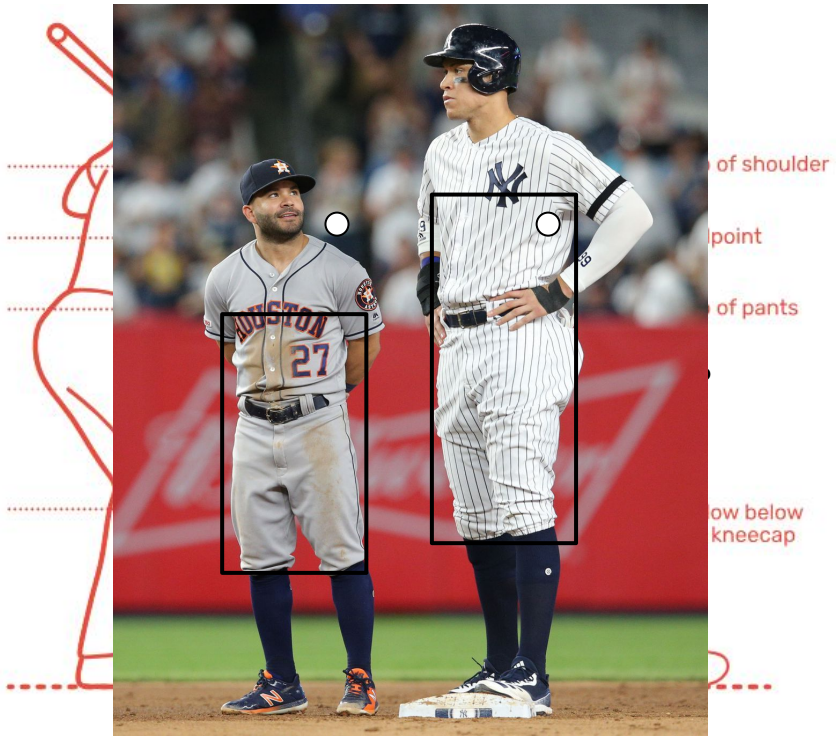


Data Loading and Preprocessing

- Z Position Normalization
  - All pZ locations were normalized based on the batter's sZ_top and sZ_bot and MLB averages

$$\frac{pZ - sZ_{bot}}{sZ_{top} - sZ_{bot}} = x$$

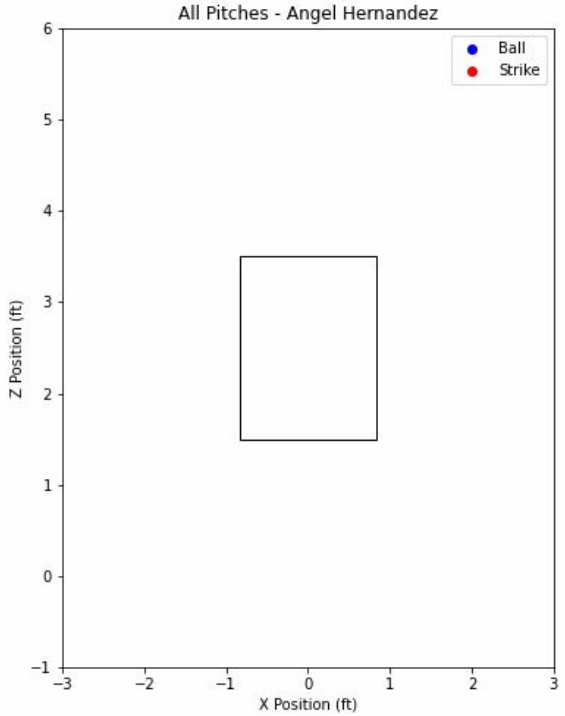
$$x(3.5 - 1.5) + 1.5 = pZ_{norm}$$



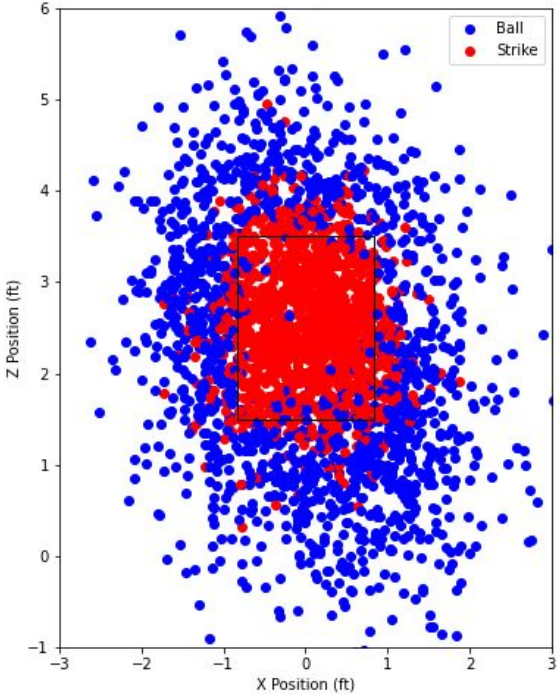
Jose Altuve vs. Aaron Judge

Data Visualization and Separation

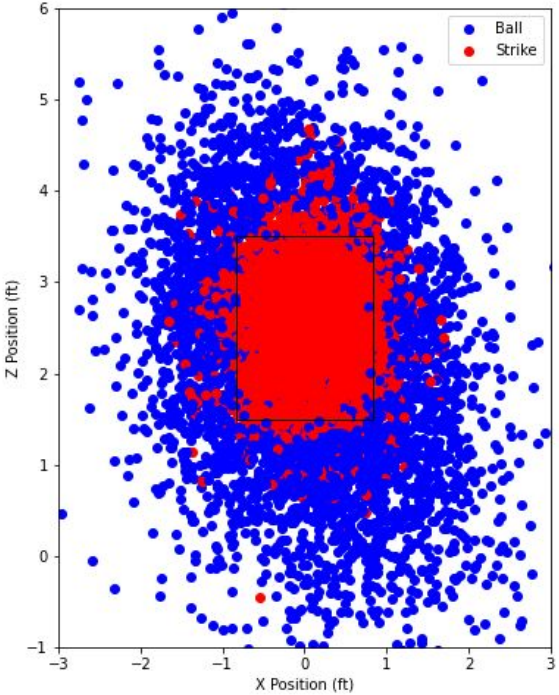
- Data Visualization
  - Pitches were converted to binary format and assigned to a color
    - 'Strike' = 1 (Red)
    - 'Ball' = 0 (Blue)
  - Strike zone was visualised using a rectangle
- Data Separation
  - Pitches were split into 95% training and 5% testing



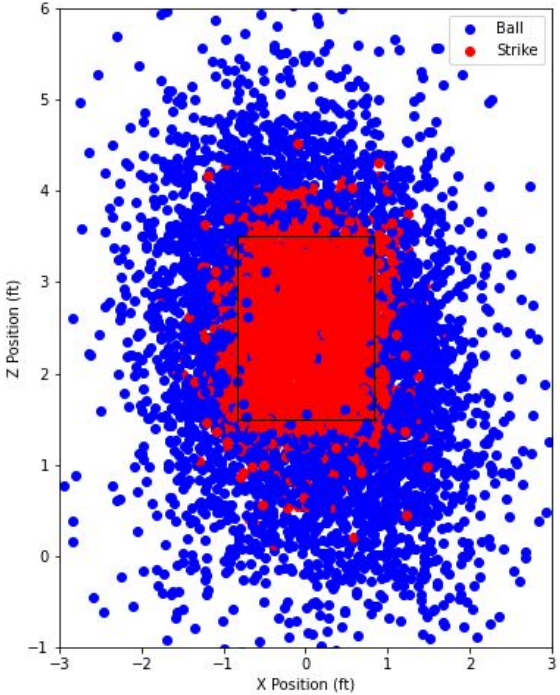
Data Visualization and Separation



Angel Hernandez

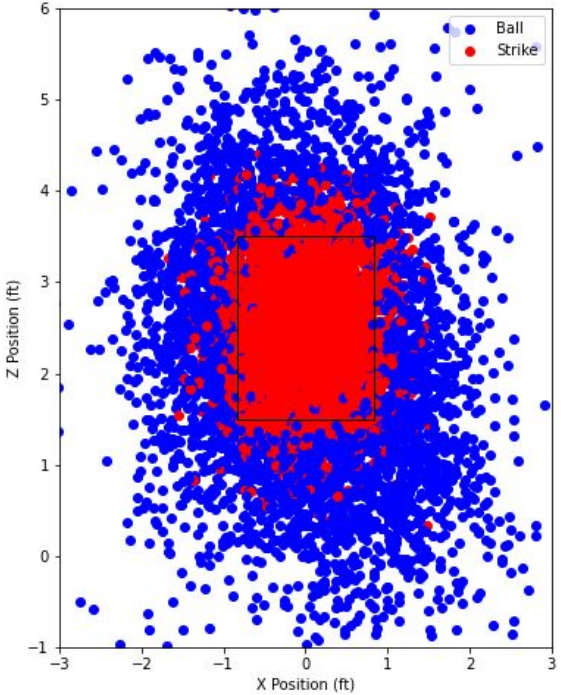


Erich Bacchus

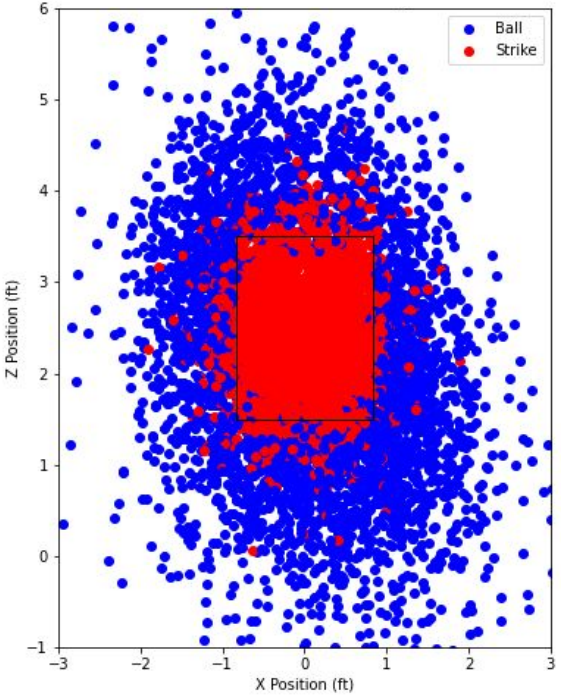


Junior Valentine

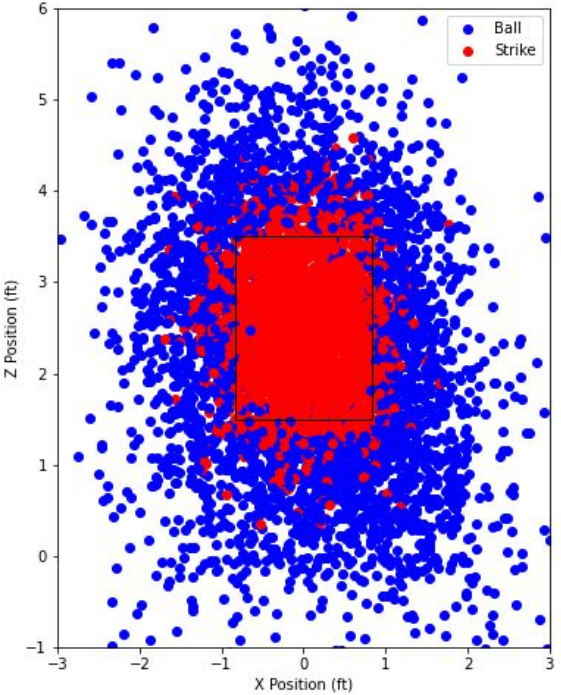
Data Visualization and Separation



Malachi Moore



Pat Hoberg



Quinn Wolcott

Method 1: Support Vector Machine

- Finds a radial hyperplane that separates the data into binary classes while attempting to maximize the margin between them
  - Predicts test pitches based on their location relative to this margin
- SVC(kernel='rbf', C=1.0, gamma='scale')
  - kernel: Radial Basis Function
  - C: Balances correct classification and maximization of the margin
  - gamma: Defines the influence of a single training example

Method 2: Gradient Boosting Classifier

- Combines predictions from multiple weak learners to create a strong predictive model
  - Series of weak learners that target the residuals of the prior model
- GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=None)
  - n_estimators: Number of boosting stages to perform
  - learning_rate: Scales the contribution of each weak learner
  - max_depth: Limits the number of nodes in the tree

Method 3: Random Forest Classifier

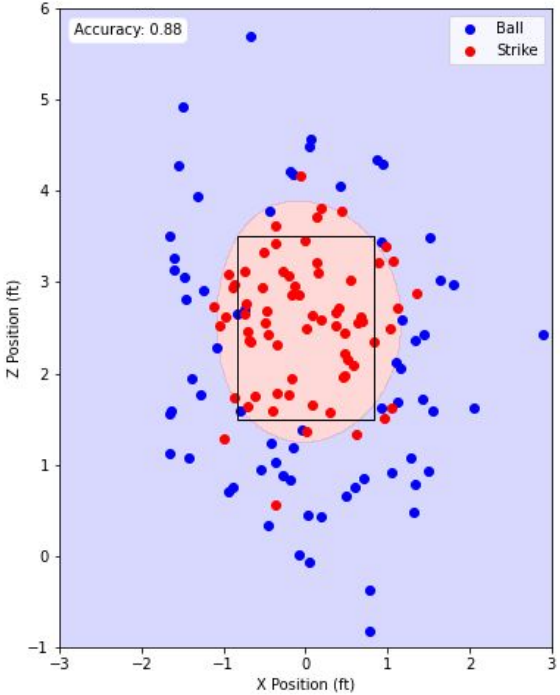
- Builds multiple decision trees using random subsets of the training data and determines predictions using a majority vote
  - Less prone to overfitting compared to individual decision trees
- RandomForestClassifier(n_estimators=200, max_depth=10, min_samples_split=10, min_samples_leaf=1)
  - n_estimators: Number of trees in the forest
  - max_depth: Maximum depth of the tree
  - min_samples_split: Number of samples required to split a node
  - min_samples_leaf: Number of samples required to be at a leaf node

Decision Boundary Visualization

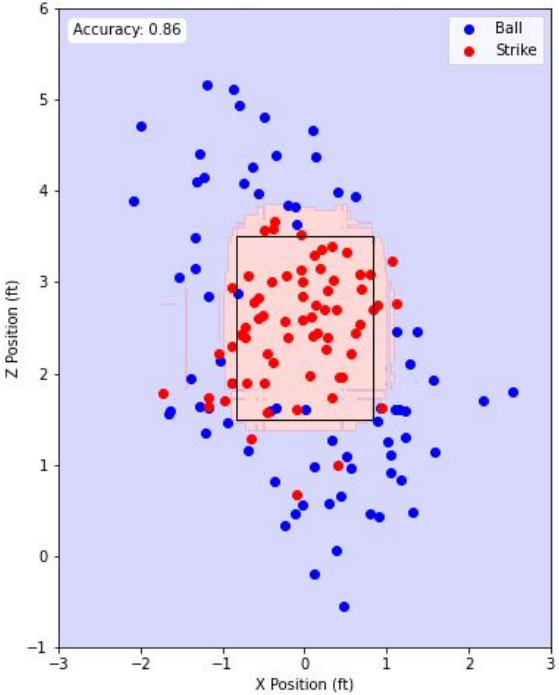
- Contour plots were used to visualize and compare the decision boundary of all three models
  - Test pitches were overlaid to easily identify discrepancies between actual and predicted calls
  - Accuracy of the model's ability to replicate the umpires calls was included on the plots

Decision Boundary Visualization

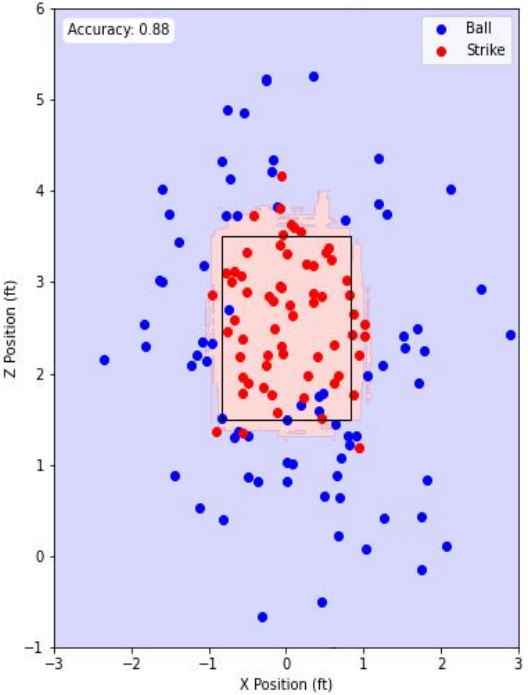
Angel Hernandez



Support Vector Machine



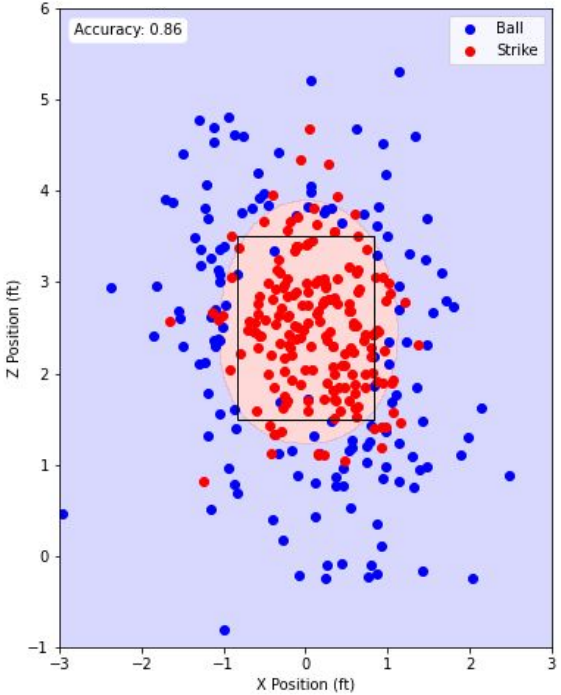
Gradient Boosting Classifier



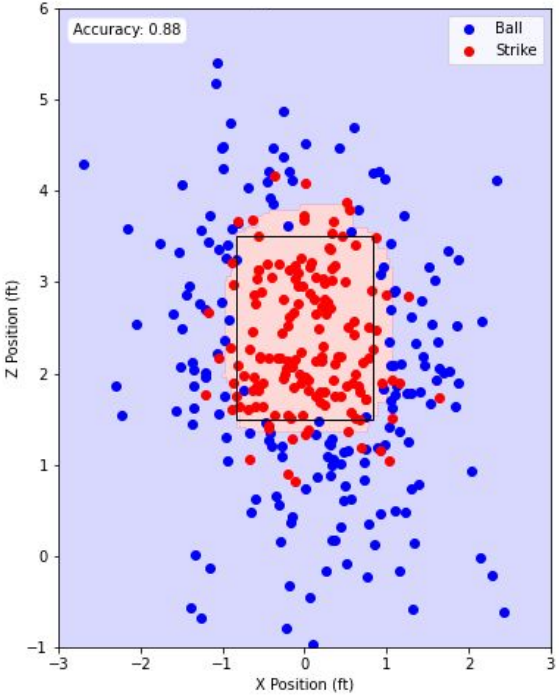
Random Forest Classifier

Decision Boundary Visualization

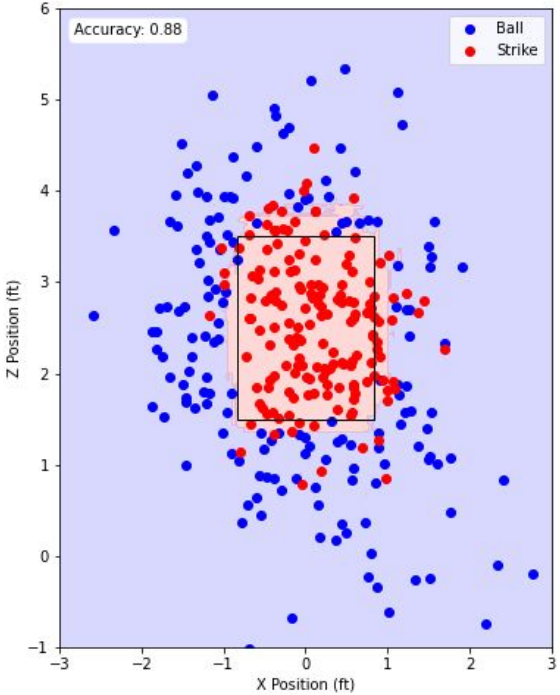
Erich Bacchus



Support Vector Machine



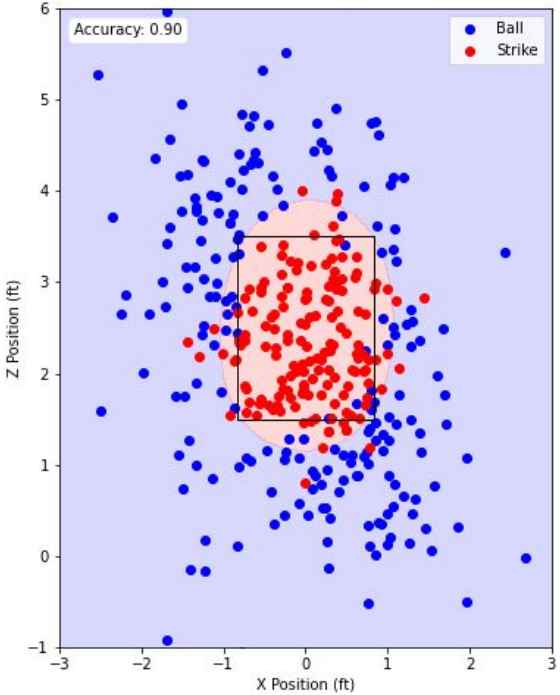
Gradient Boosting Classifier



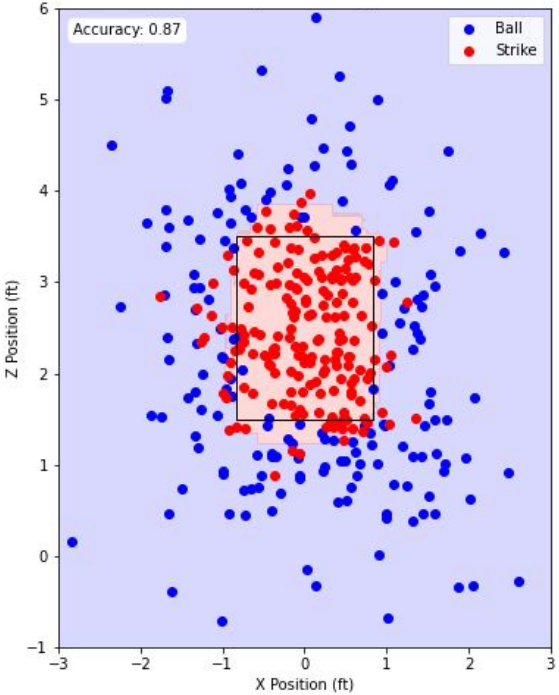
Random Forest Classifier

Decision Boundary Visualization

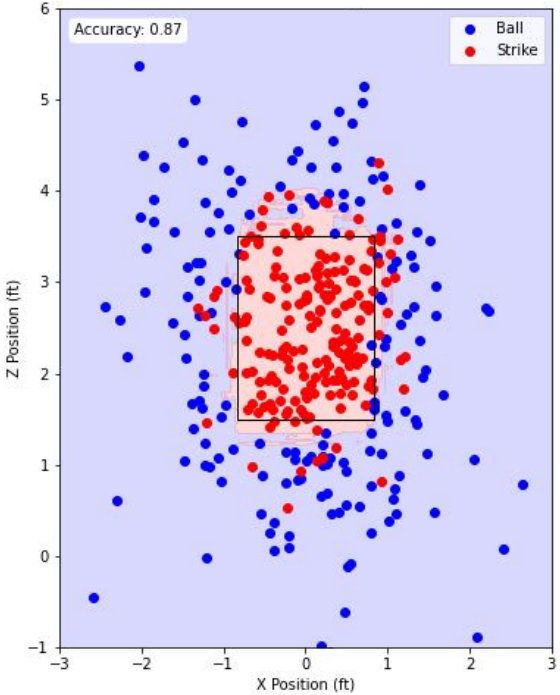
Junior Valentine



Support Vector Machine



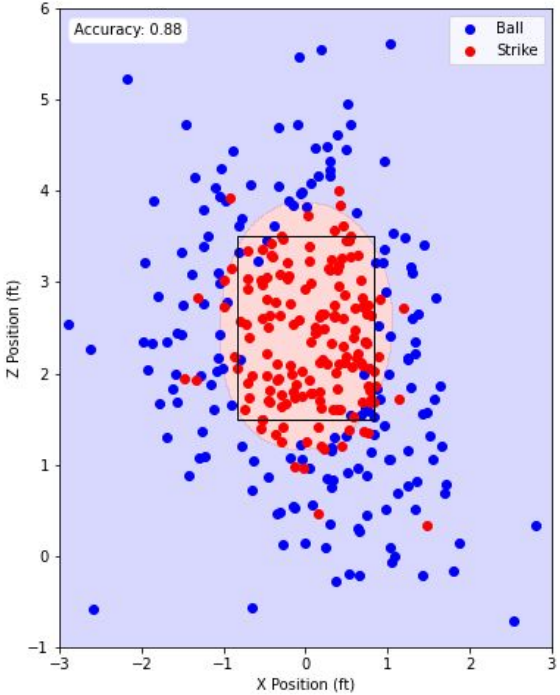
Gradient Boosting Classifier



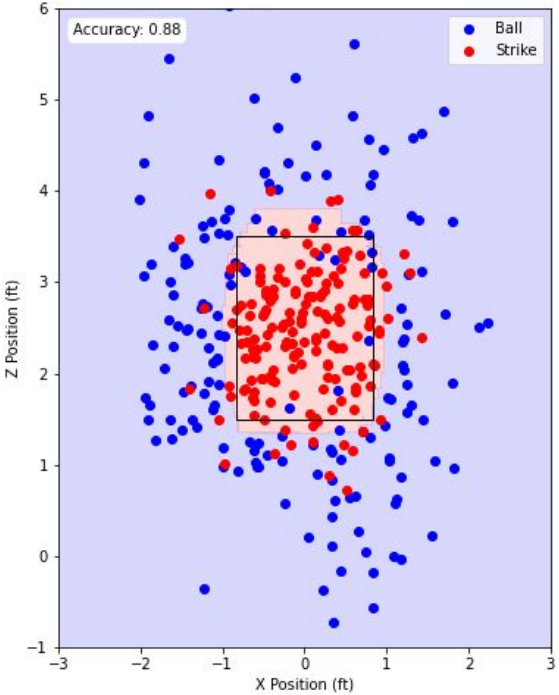
Random Forest Classifier

Decision Boundary Visualization

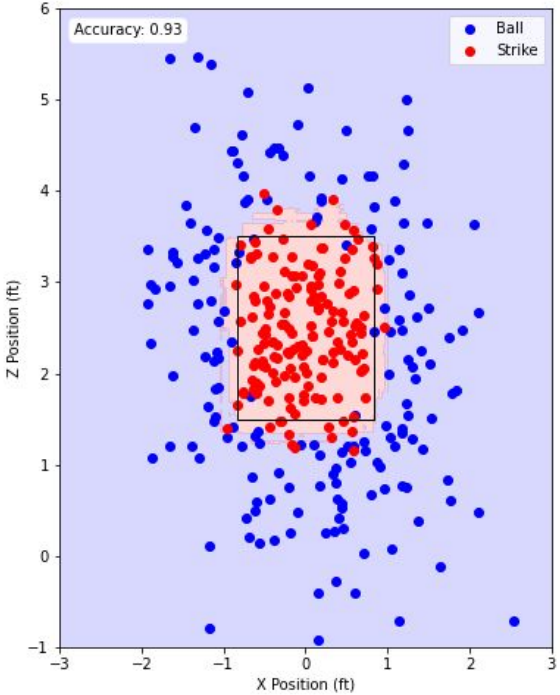
Malachi Moore



Support Vector Machine



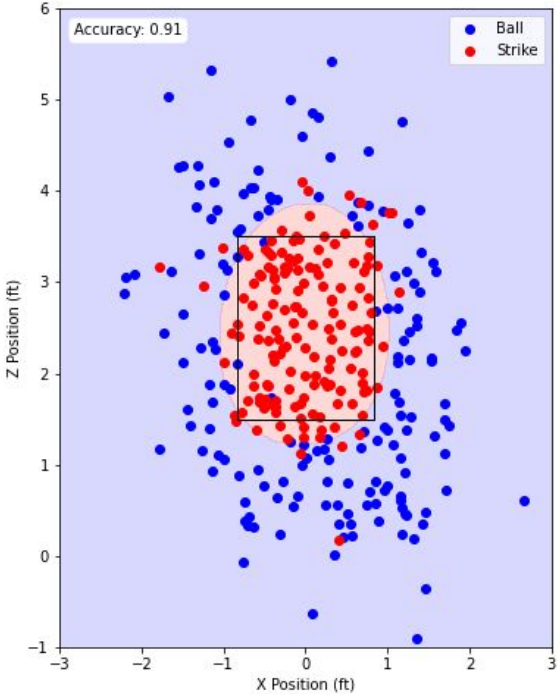
Gradient Boosting Classifier



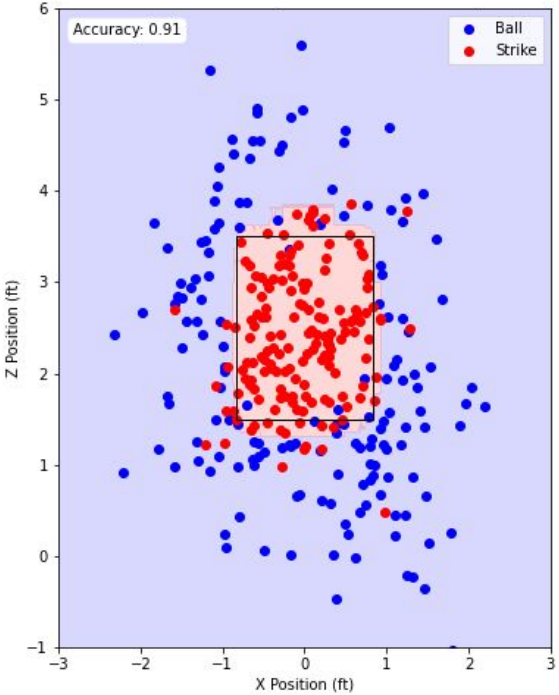
Random Forest Classifier

Decision Boundary Visualization

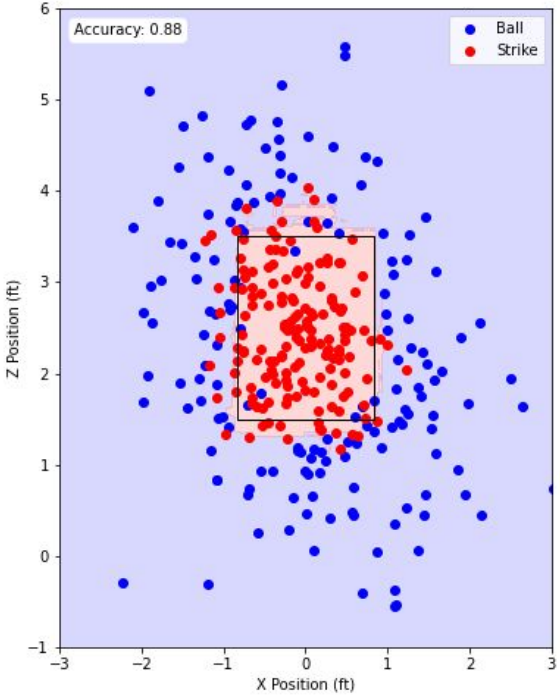
Pat Hoberg



Support Vector Machine



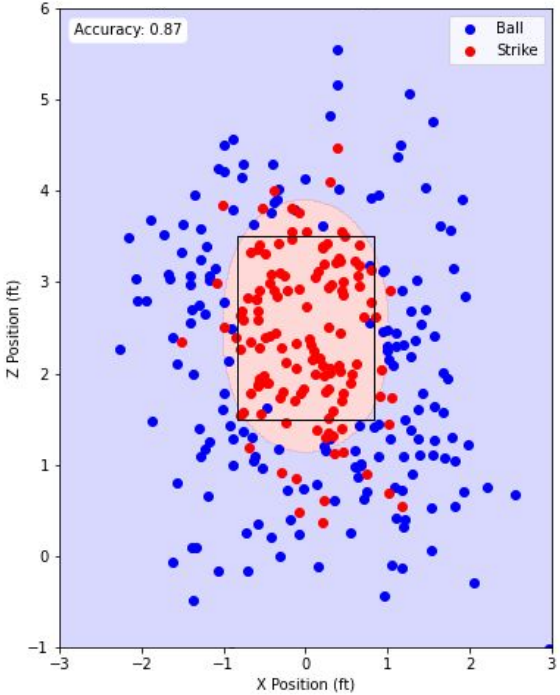
Gradient Boosting Classifier



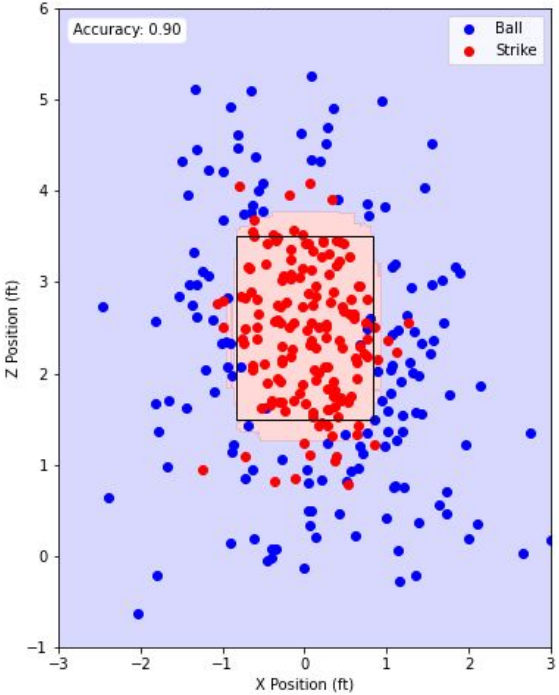
Random Forest Classifier

Decision Boundary Visualization

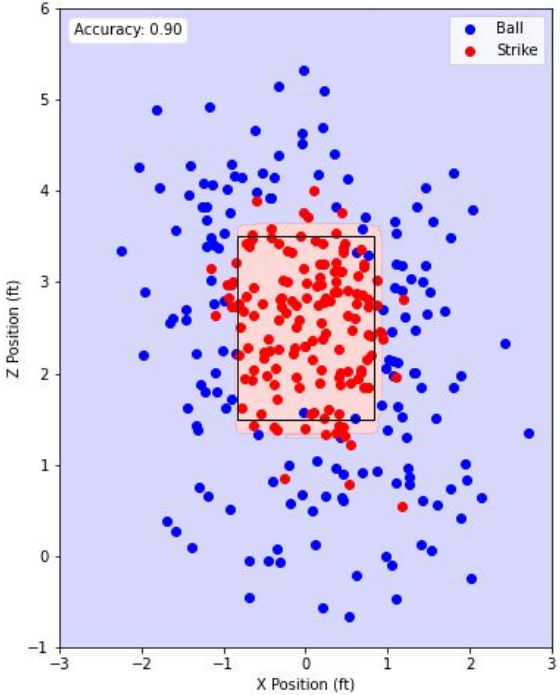
Quinn Wolcott



Support Vector Machine



Gradient Boosting Classifier



Random Forest Classifier

Optimization

- Hyperparameter Tuning
  - GridSearchCV was used to find the optimal hyperparameters for the random forest classifier
  - Helped create a more optimized model and achieve better performance.
  - param_grid = {'n_estimators': [50, 100, 200],
    'max_depth': [5, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5]}

Optimization Results

	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 models all have a similar average accuracy around 0.9.

	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**

Future Applications

- “What-If” Scenarios
  - Use the models to test how other umpires would have handled controversial or close calls
- Game Simulations
  - Use the model to call a complete game and determine if there would have been a different outcome
- Zone “Opening”
  - Use the model to support whether an umpire widened their personal zone in an attempt to end a game

Conclusion

1. Pitch strike zone data was collected and normalized for a machine learning analysis focus of the “human effect” of umpires.

2. The data was visualized based on binary classification and split into 95% training and 5% testing subsets.

3. Support vector machine, gradient boosting classifier, and random forest classifier models were developed for 6 major league umpires.

4. Hyperparameter tuning was implemented into the random forest classifier and all models had an accuracy around 0.9.

5. Identified possible future applications for umpire strike zone models.

MECHANICAL ENGINEERING
TEXAS A&M UNIVERSITY

Thank You!

Questions?

References

[1] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html