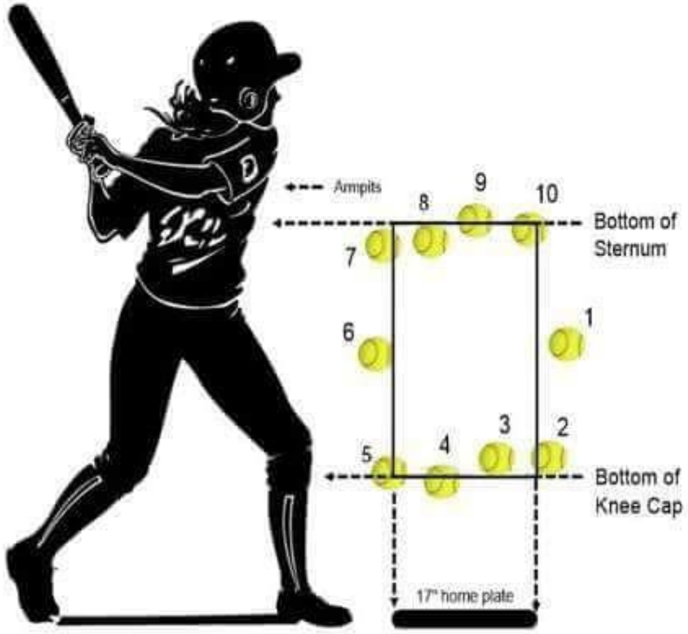
Catcher Framing Model

LSU Softball – Trackman Data

What is Catcher Framing?

- The technique of receiving a pitch in a way that convinces the Umpire to call a borderline pitch a strike
  - Borderline: on or around the strike zone
  - The goal is to catch the ball in a certain way, that makes it look like a strike has been thrown
  - Balls= pitches that cross of the home plate outside of the strike zone
  - Strikes= pitches thrown that cross the home plate in the strike zone
  - 3 strikes= 1 Out (which benefits the defense - LSU in the given situation)
  - 4 Balls= A walk (not wanted)



Rulings: 1) ball, 2) strike, 3) strike, 4) ball, 5) ball, 6) strike, 7) strike, 8) strike, 9) ball, 10) ball

- By using subtle, controlled glove movements to "bring the ball back" into the strike zone, catchers aim to convert potential balls into called strikes.

Problem

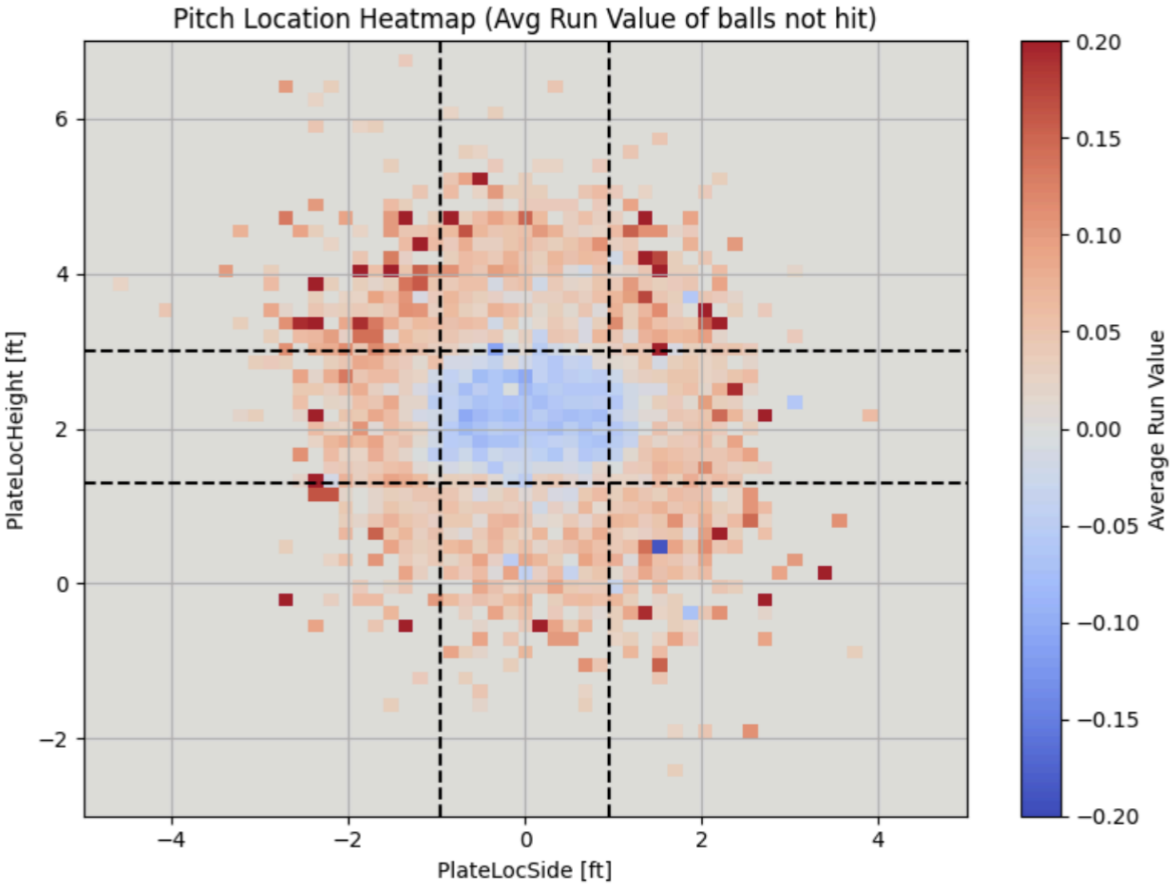
- We want to improve catcher framing by analyzing the ball's location in terms of the strike zone, in comparison to called strikes.
- How can the LSU catcher best frame a pitch to convince the Umpire it was a strike?
  - The art of turning borderline pitches into favorable calls

What We Did

- Made different models to illustrate the region around the strike zone, better known as the "Shadow Zone"
  - The Shadow Zone is classified as the area surrounding the strike zone, atone balls width
  - This zone is where balls are most likely to be called strikes
- We also wanted to determine the error rate of the Umpire for any given day

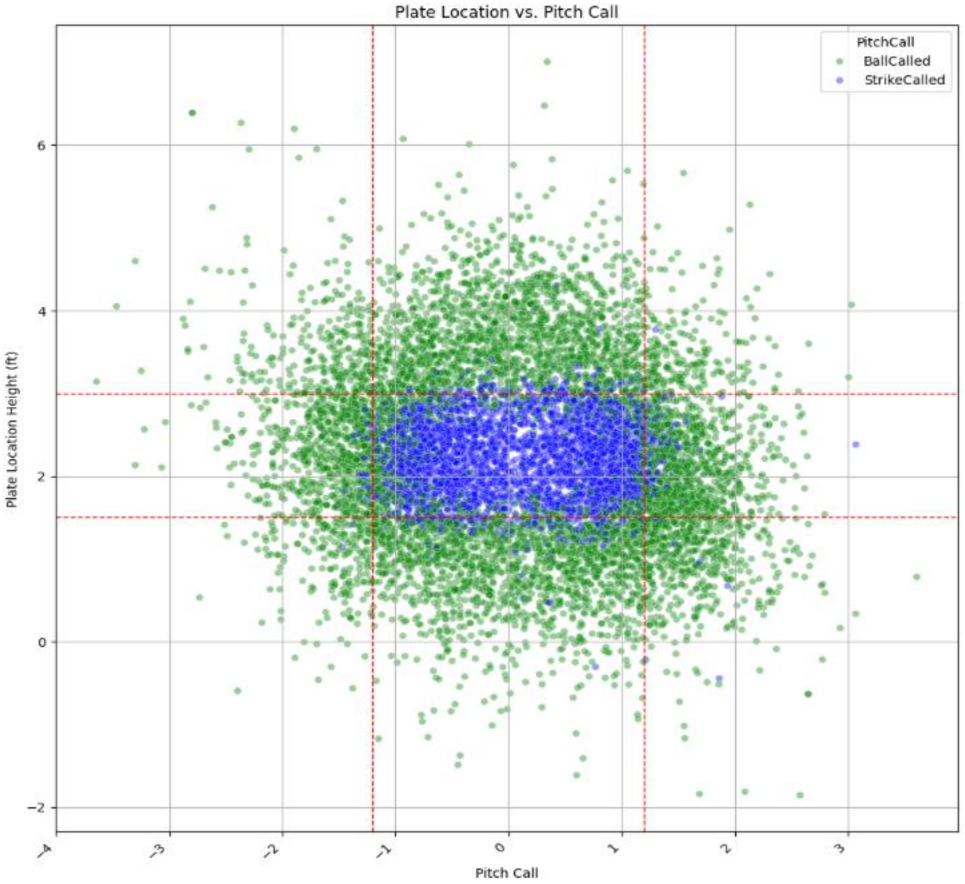
Previous Data

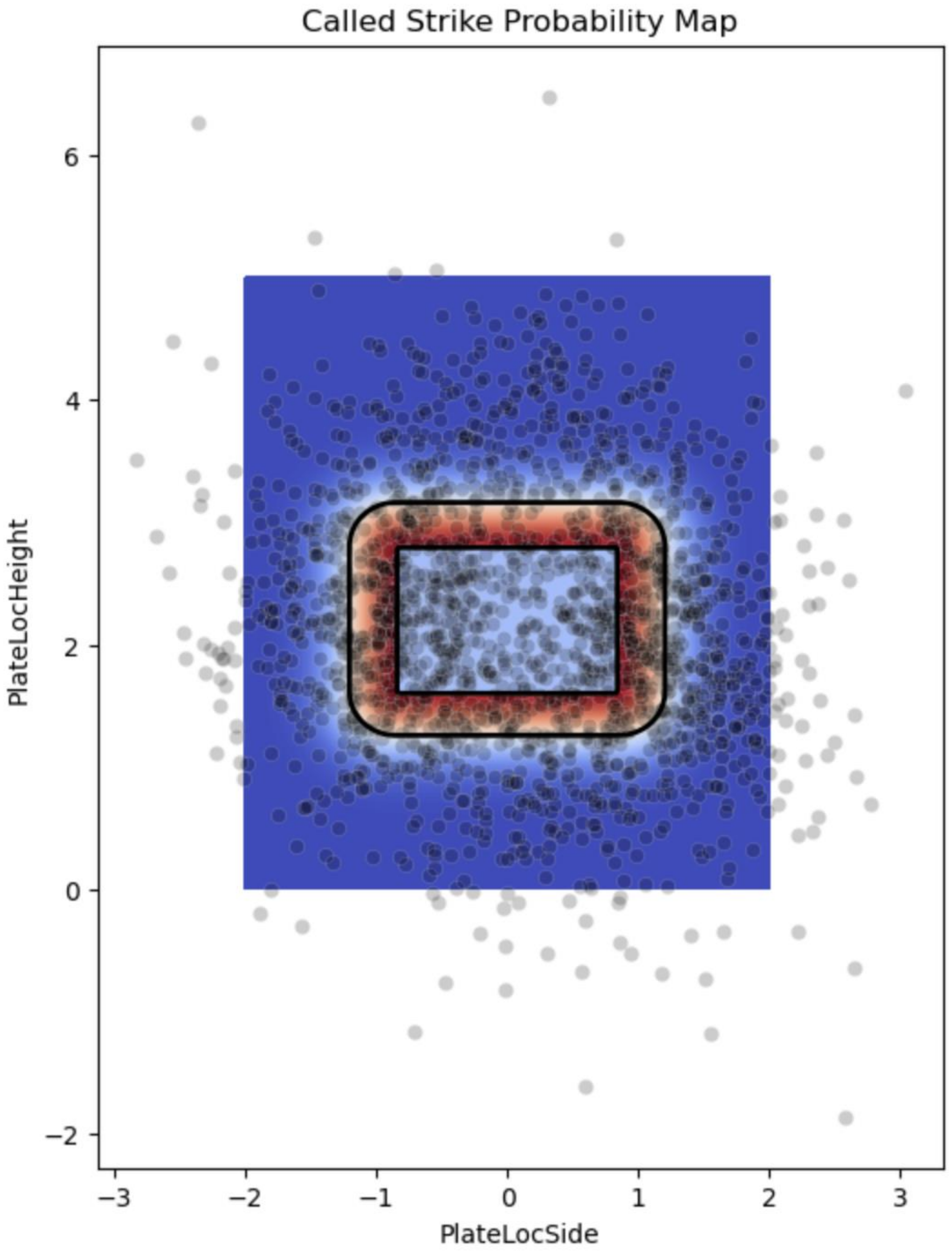
- The colors represent the average run value when a ball is not hit, and each unit square represents the location of a pitch
- In the blue zone, the probability of a batter hitting is higher, and in the red zone, it is lower
- Pitches blue zone are more likely to be called strikes, and the red are more likely to be called balls



Strike Zone Model

- The model from last semester shows the strike zone, we decided to adjust it using a linear regression model to show the zones relative to balls and strikes called
- The strike zone we used was based on the general zone of 3ft by 2.4ft (given to use by Dr. Jermain)
    - The red lines represent the strike zone which we found to be 3ft by 1.5ft
- Blue represents strikes called
- Green represents balls called





Shadow Zone

- We created a strike probability map highlighting the shadow zone, falling in between the black lines
- The strike zone is the space inside the center box
- The red represents the heart zone, and the orange represents the shadow zone
- The light blue outside the black line represents the chase zone, and the dark blue represents the waste

Machine Learning Models for Baseball Strike-Call Prediction

Three supervised learning models were tested:

- **Random Forest**
- **Logistic Regression**
- **Linear Regression**

Evaluation Metrics

- Visualization of strike probabilities
- Error Rate
- Accuracy
- Precision
- ROC-AUC

Why We Chose These Three Models

We selected three models to compare **different levels of complexity, interpretability, and predictive power** for strike-call classification.

1. Logistic Regression

- Standard benchmark for binary classification problems
- Easy to interpret probabilities and feature effects

2. Linear Regression

- Included as a simple statistical benchmark
- Helps demonstrate why regression methods are weaker for binary outcomes

3. Random Forest

- Powerful nonlinear machine learning model
- Captures complex strike-zone boundaries and interactions

Why Compare All Three?

By comparing simple linear models with an advanced nonlinear model, we can determine whether strike-call prediction requires more sophisticated machine learning methods.

Why We Used These Performance Metrics

Accuracy

- Measures the percentage of total predictions that were correct.
- Useful for understanding overall model performance.

Precision

- Measures how often predicted strikes were actually strikes.
- Important because false strike calls can strongly affect game outcomes.

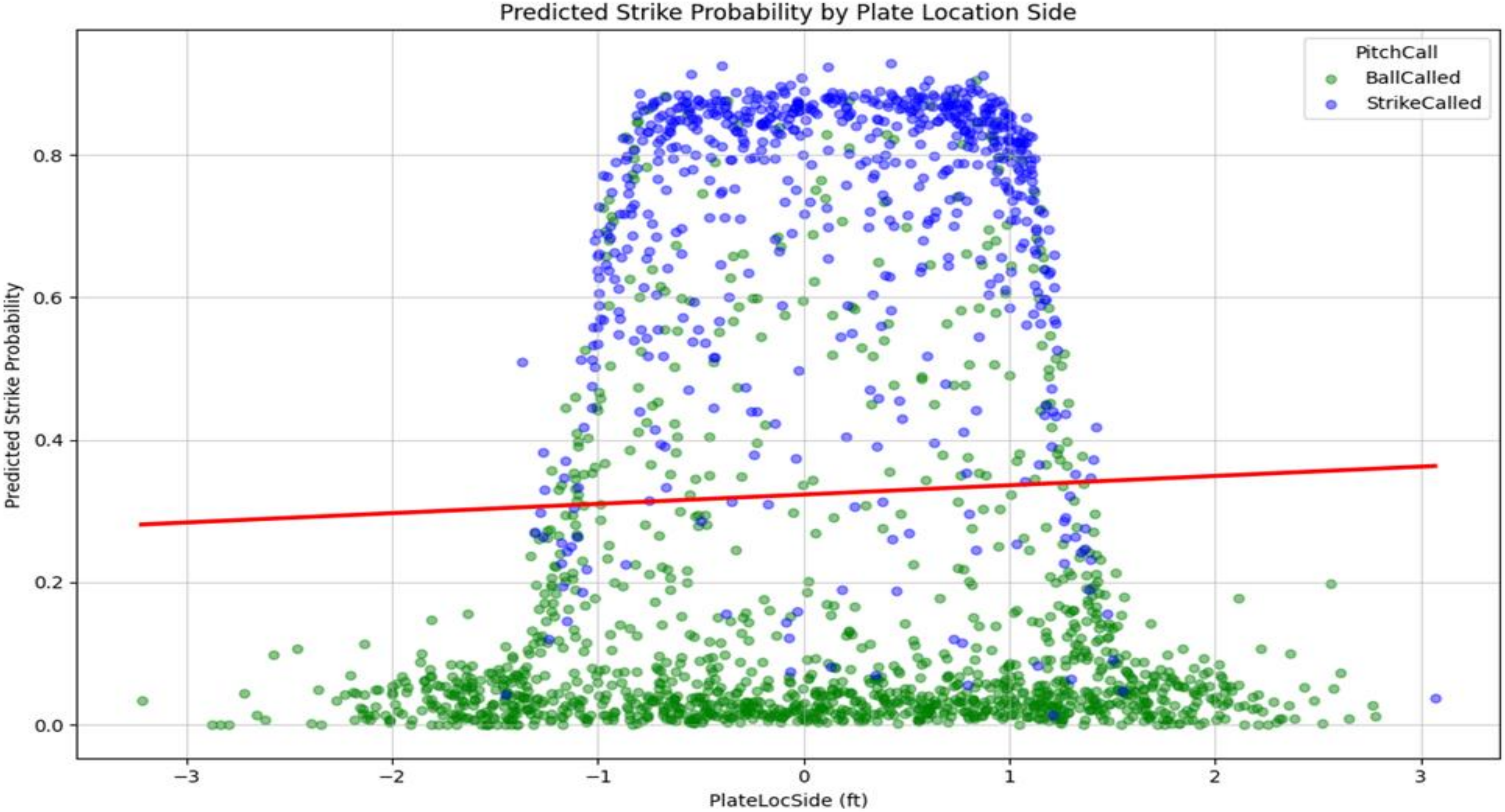
ROC-AUC

- Measures how well the model separates strikes from balls across all thresholds.
- Higher values indicate stronger classification ability.

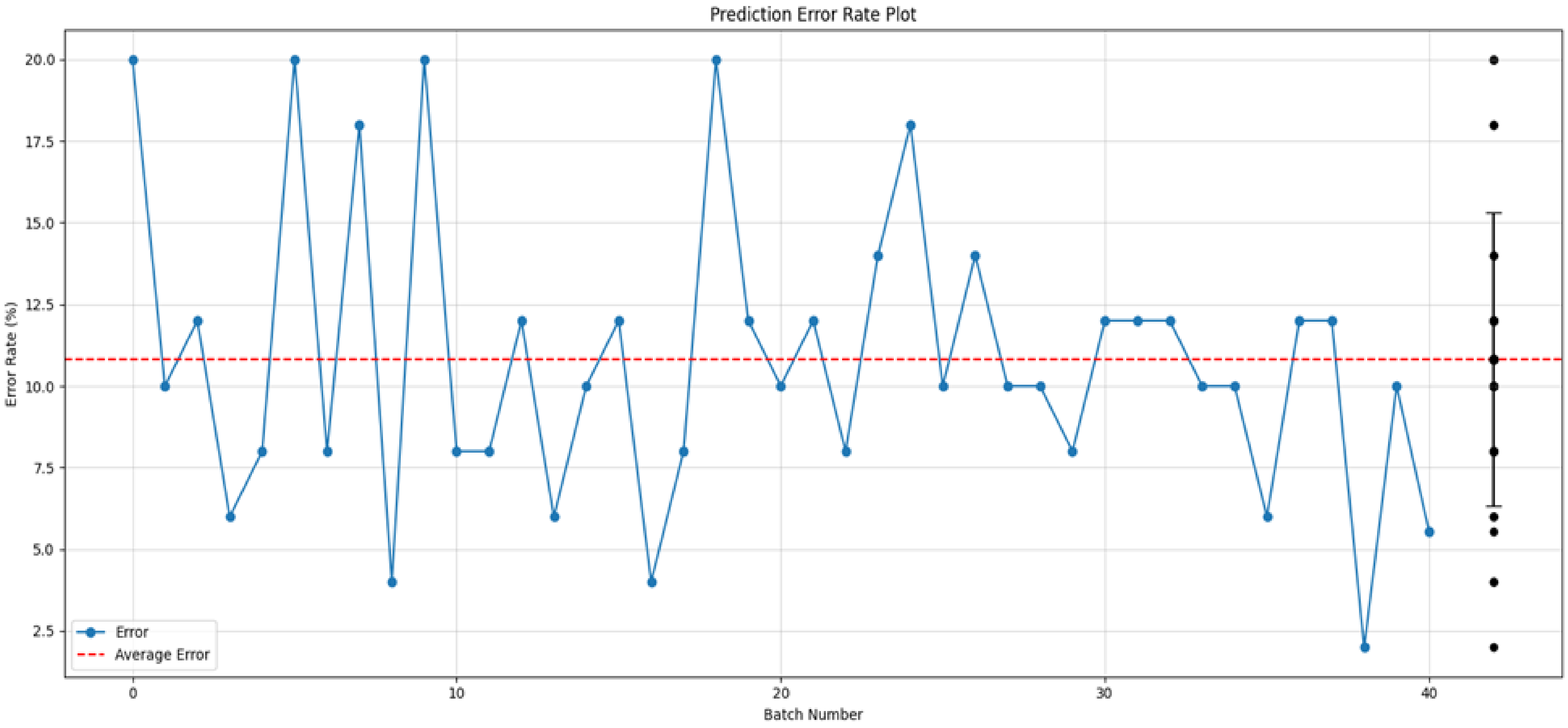
Why We Chose These Metrics

- They are standard and widely accepted for **binary classification problems**.
- They evaluate overall correctness, prediction reliability, and class separation.

Random Forest Model Results



Random Forest Model Results



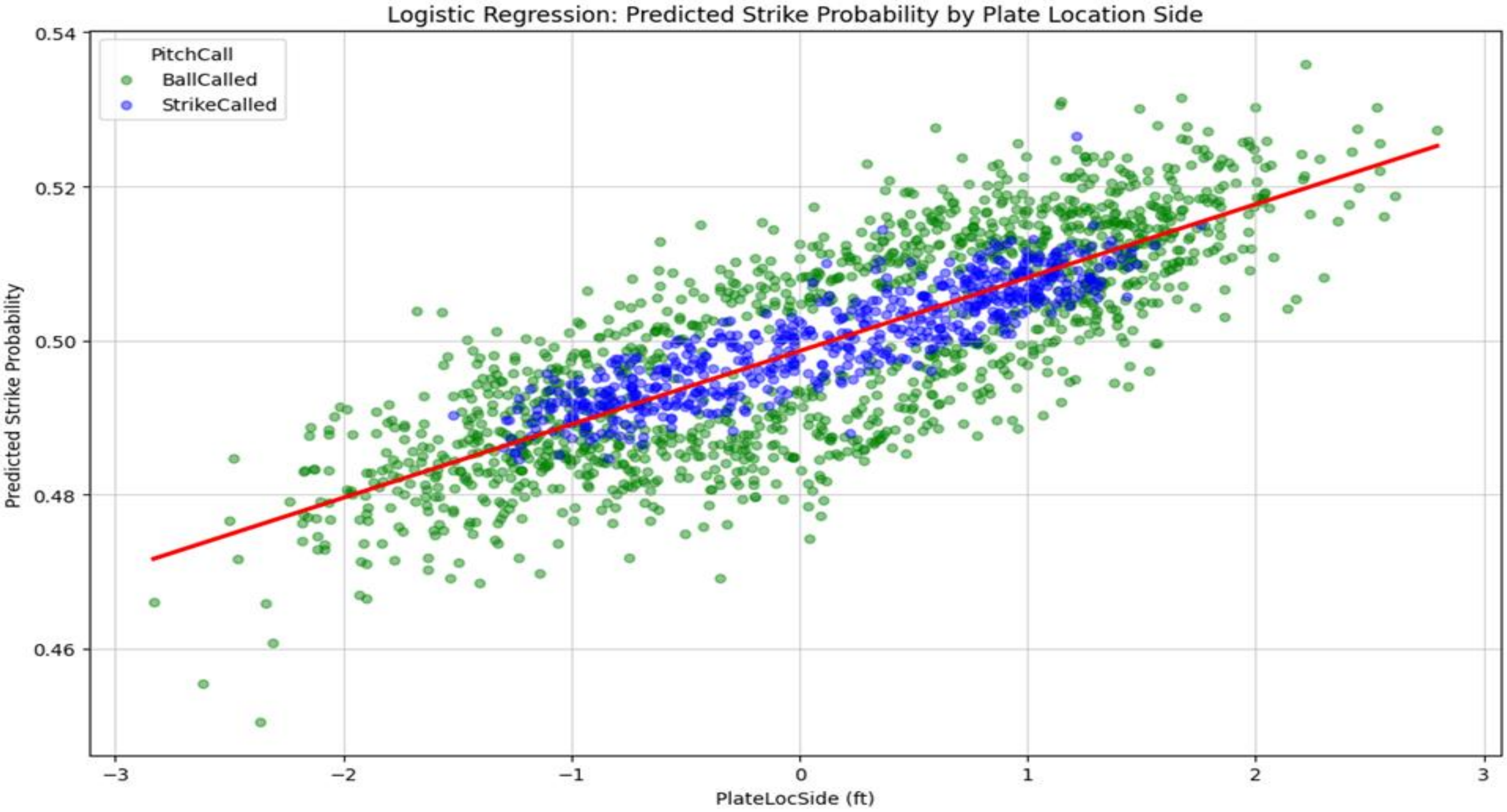
Random Forest Model Results

- Performance Metrics

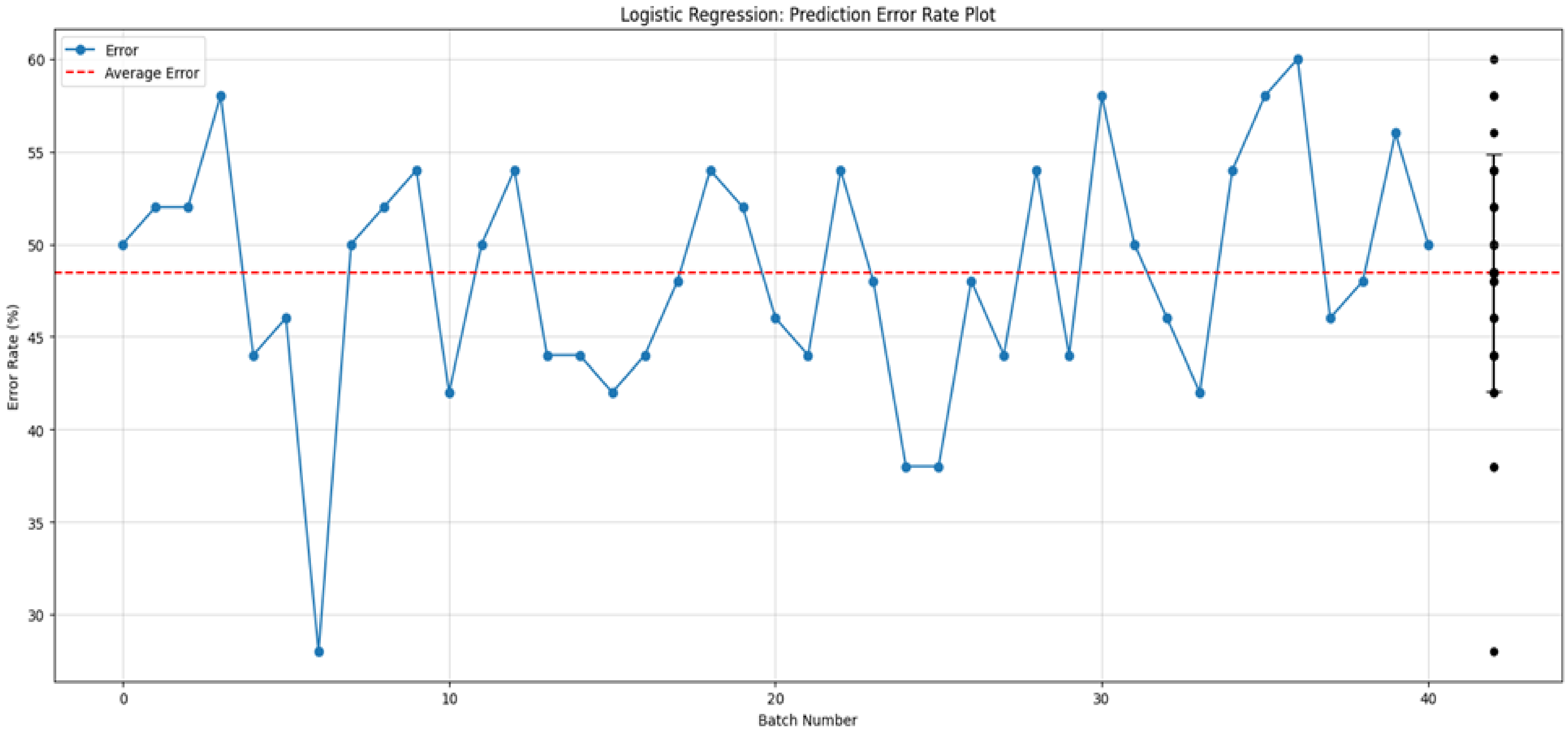
Metric	Value
Accuracy	89.1%
Precision	82.7%
ROC AUC	0.9545

- Captured nonlinear strike-zone patterns effectively
- Stable average error rate around **11%**
- Strong overall model performance

Logistic Regression Model Results



Logistic Regression Model Results



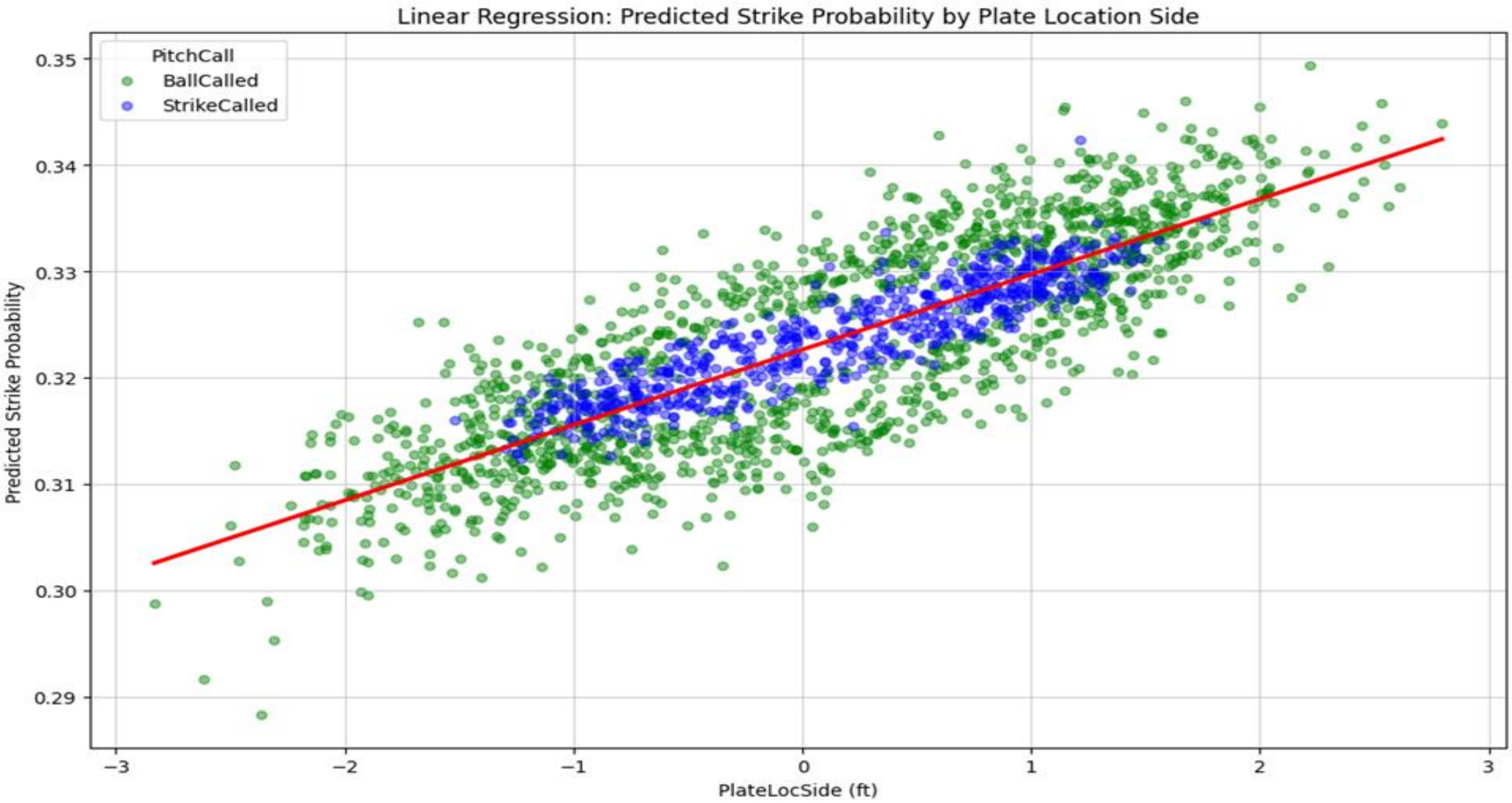
Logistic Regression Model Results

- Performance Metrics

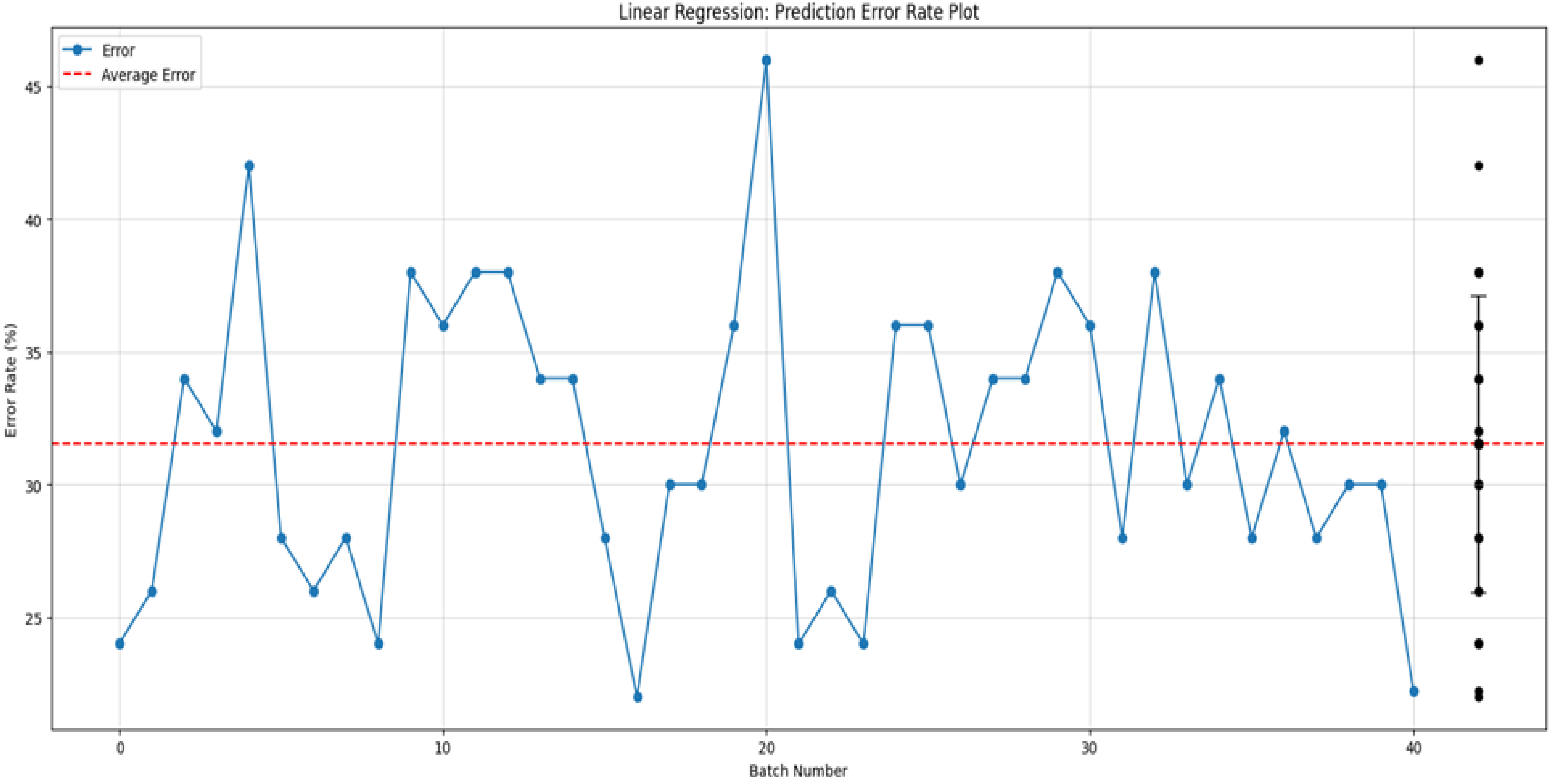
Metric	Value
Accuracy	51.6%
Precision	33.2%
ROC AUC	0.5164

- Weak strike/ball separation
- Predictions concentrated near 0.50
- Linear boundary too simplistic

Linear Regression Model Results



Linear Regression Model Results



Linear Regression Model Results

- Performance Metrics

Metric	Value
Accuracy	68.3%
Precision	0.0%
ROC AUC	0.5165

- Failed to classify strike outcomes effectively
- Poor separation between strikes and balls
- Not suitable for binary classification

Model Comparison Summary

- Performance Metrics

Model	Accuracy	Precision	ROC AUC
Random Forest	89.1%	82.7%	0.9545
Logistic Regression	51.6%	33.2%	0.5164
Linear Regression	68.3%	0%	0.5165

- Random Forest is the best model

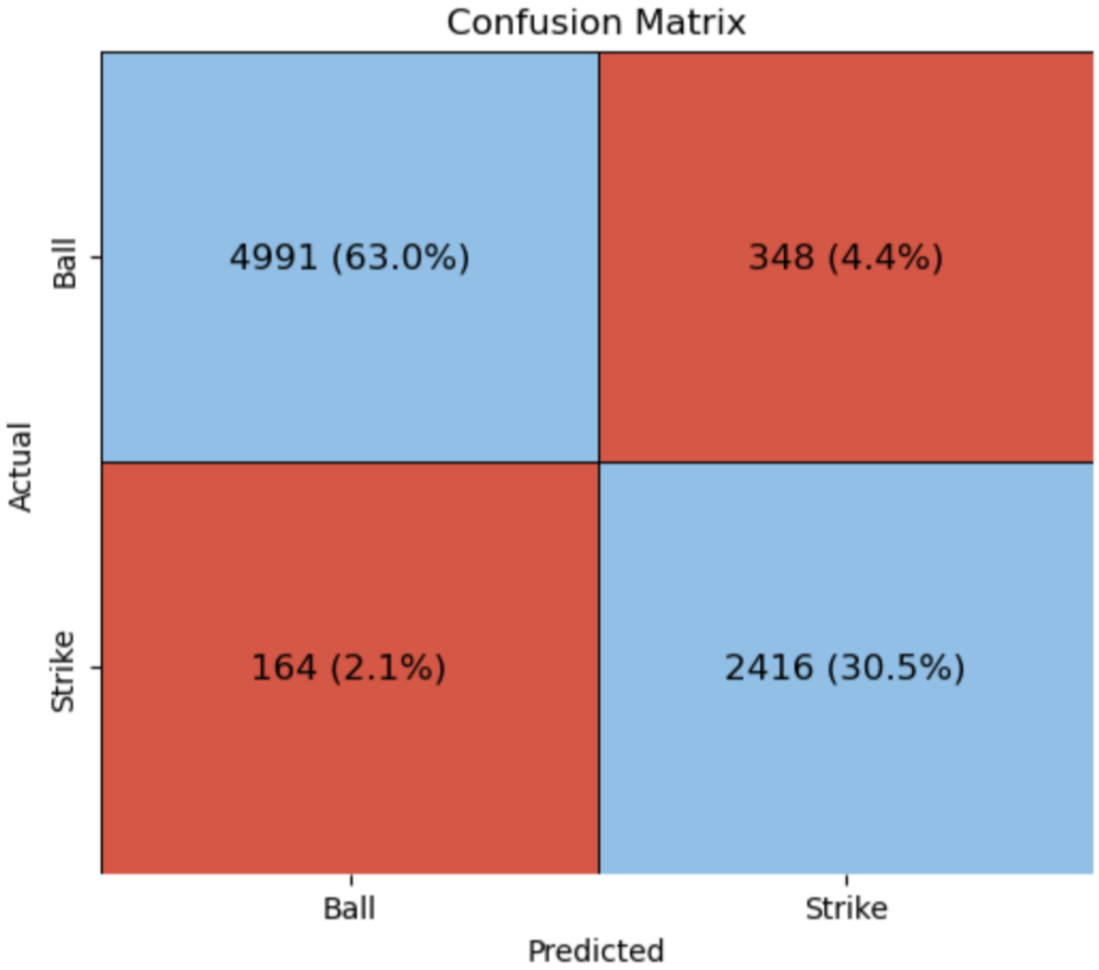
Final Thoughts

- Random Forest delivered the best predictive performance.
- Logistic Regression served as an interpretable baseline but struggled.
- Linear Regression was not appropriate for this binary task.
- Strike-zone prediction benefits from models that learn nonlinear relationships.

Final Conclusion

- **Random Forest is the most effective model for strike-call prediction in this dataset.**

Random Forest Model Confusion Matrix



RFM: Framing Value

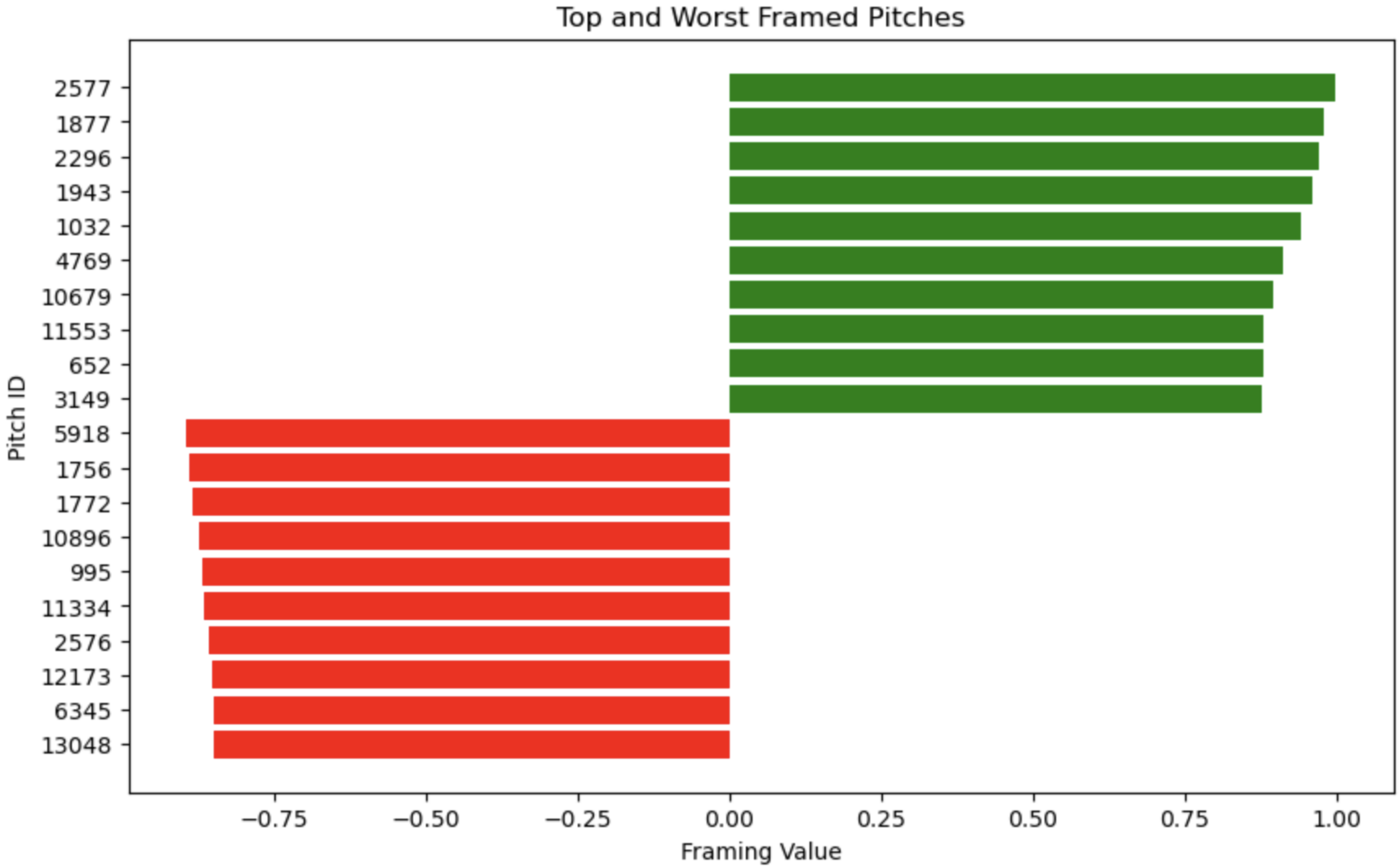
- We used a metric called Framing Value in order to isolate the catcher's impact on each ball, using the formula:

Framing Value = Actual Outcome - Expected Strike Probability

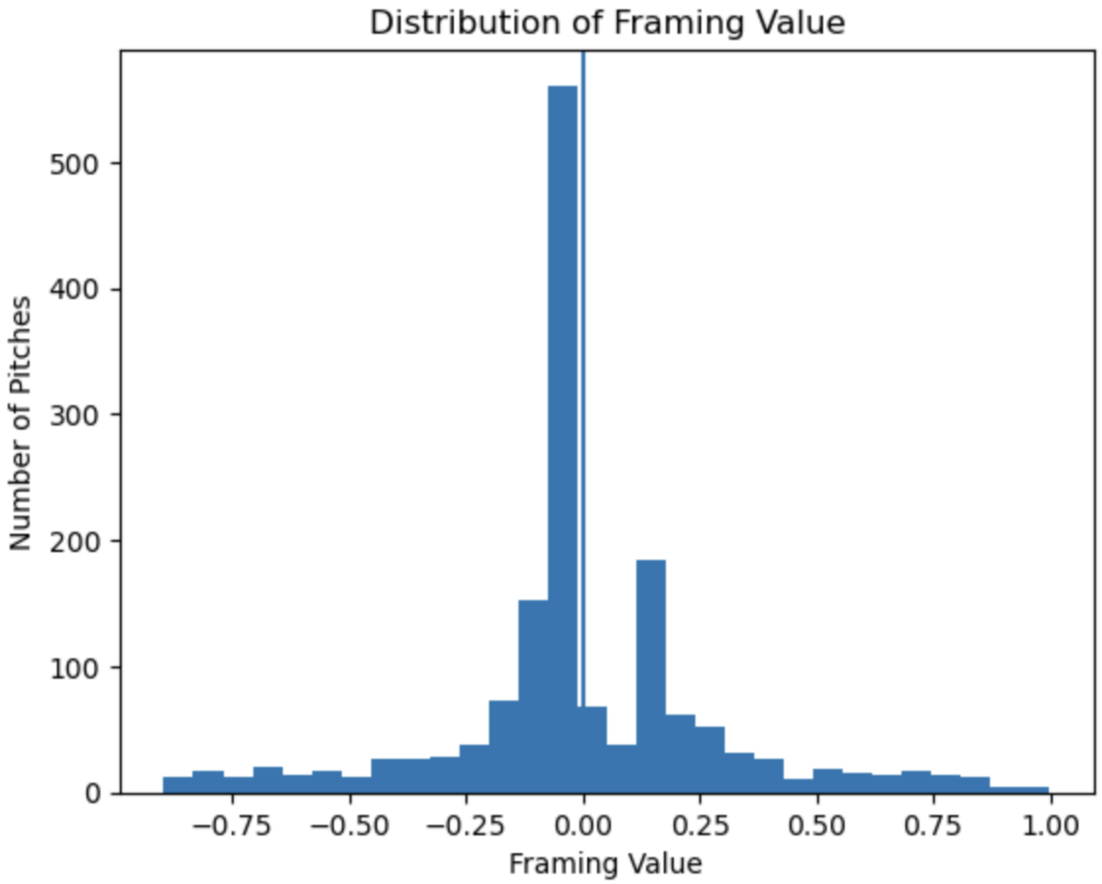
where Actual Outcome is 1 for a strike and 0 for a ball.

- The positive values represent good framing, where the catcher was able to "steal" a strike.
    - Occurs when a pitch is called a strike despite having a low expected probability.
    - For example, a framing value near +0.9 means a pitch only had a 10% chance of being a strike, but the catcher framed it well enough to get the call
- The negative values occur when a pitch is called a ball despite having a high probability of being called strike
    - Likely due to poor framing or missed location

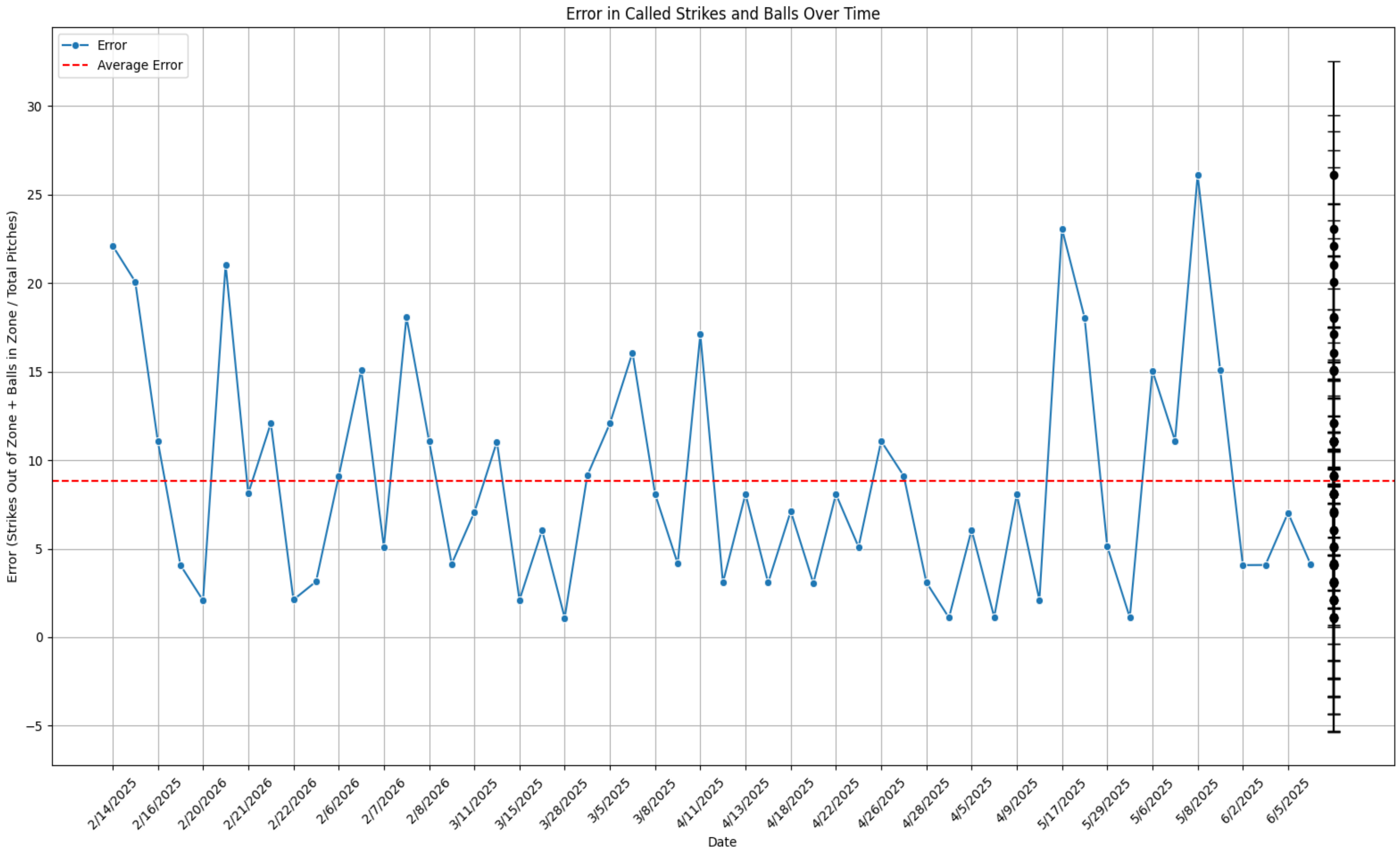
Random Forest Model Results



Framing Value Distribution



- The Framing Value distribution demonstrates how pitch outcomes generally align with the expected calls.
- The bulk of pitches have a framing value of around 0.0, meaning that the catcher had little to do with the call being strike or ball.
    - This indicates that most pitches are called exactly as expected based on their location.
- The outliers (tails) more clearly represent distinct catcher performance.



Error = "StrikesOutZone" + "BallsInZone" / "TotalPitches"

Umpire Error Rate Model

- To estimate umpire reliability, we took the error per game to see the worst umpire calls and the best
- Our mean/average line sits at 8.809
- With a standard deviation of 6.436 represented with the error bar

Other Models That Could Be Used

- 1. Neural Networks
- Can learn highly complex pitch-call patterns
- Useful with large datasets and many variables
- 2. K-Nearest Neighbors (KNN)
- Classifies pitches based on nearby historical pitches
- Simple but computationally slower for large datasets
- 3. Decision Tree
- Easy to visualize and interpret
- Simpler version of Random Forest
- Testing boosting models and neural networks may further improve strike-callprediction accuracy beyond Random Forest.

Suggested Next Steps

- Use the **Random Forest model** and error-rate analysis to identify pitch locations where called strikes are most likely to be gained through catcher framing.
- Detect zones where umpires frequently call strikes on borderline pitches and where balls are commonly misclassified.
- Provide catchers with data-driven training maps showing the most valuable locations to receive pitches cleanly.
- Expand the model by adding more features such as pitch type, count, batter stance, pitcher handedness, and catcher identity.
- Compare model predictions across multiple seasons to study consistency and long-term trends.
- Apply the framework to support coaching decisions, player development, and game strategy.

Sources

- https://beanumber.github.io/abdwr3e/07-framing.html
- https://medium.com/@kaifranke3/strike-probability-model-and-catcher-framing-using-random-forest-7b482d6f159f

Thank you!!

Questions/Comments/Concerns?