The Art of Catcher Framing

Briana Cox, Koby Guidroz, Gretta Ek, Sachin Dahiya

Math 4997

Dr. Nadejda Drenska

April 2026

Introduction

The project we worked on was in collaboration with the LSU softball team to help generate a catcher framing model, to show the optimal zone for a catcher to frame a ball in a certain way to get a strike called, even when a ball is outside of the strike zone. This zone is deemed to be called the “shadow zone” and is typically known to be one balls width outside the strike zone. Based on the data provided, we tried to create a visual representation of the shadow zone, which will show where pitchers should aim for, where catchers should focus their attention on framing, and help train batters to better hit pitches within this zone. We also made a model to show the error rate of an Umpire for any given game. This is helpful because it shows the average error that will occur in calls during a game

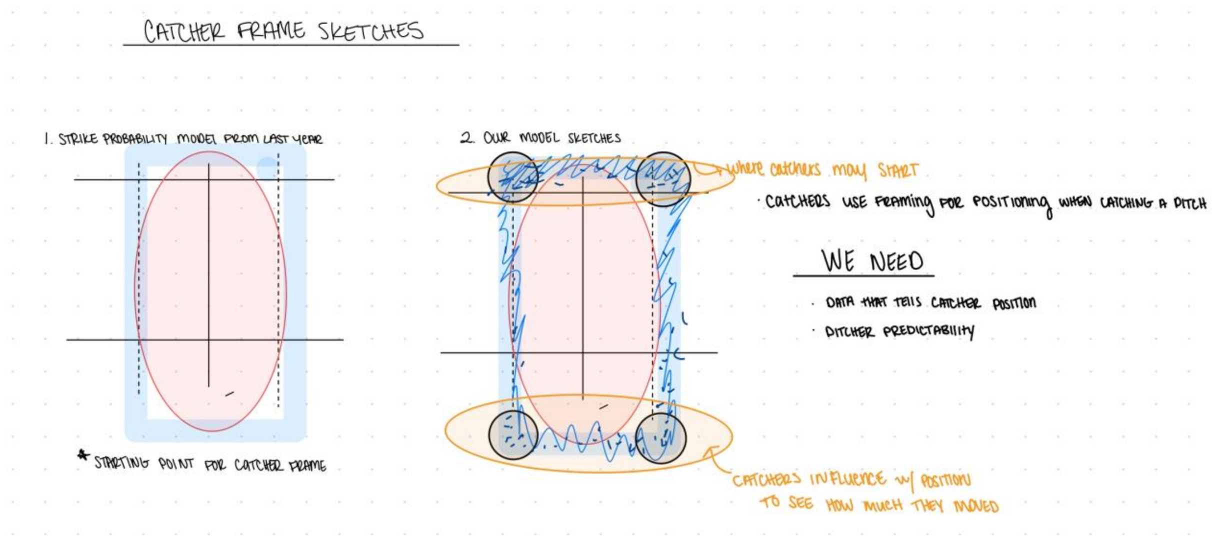
The catcher frame is also important for umpire reliability. Due to the umpire being able to make a call of ball or strike sometimes a ball will be called a strike, and a strike will be called a ball. So, not only did we aim to find the catcher frame but also to see the error per game. Which would determine in what game had the best umpire who stayed true to the zone but also the worst umpire who had a significant error in those games. It is important for the coaching staff to see these results as they can better prepare the softball team for how much they need to focus on their frame.

Data collection

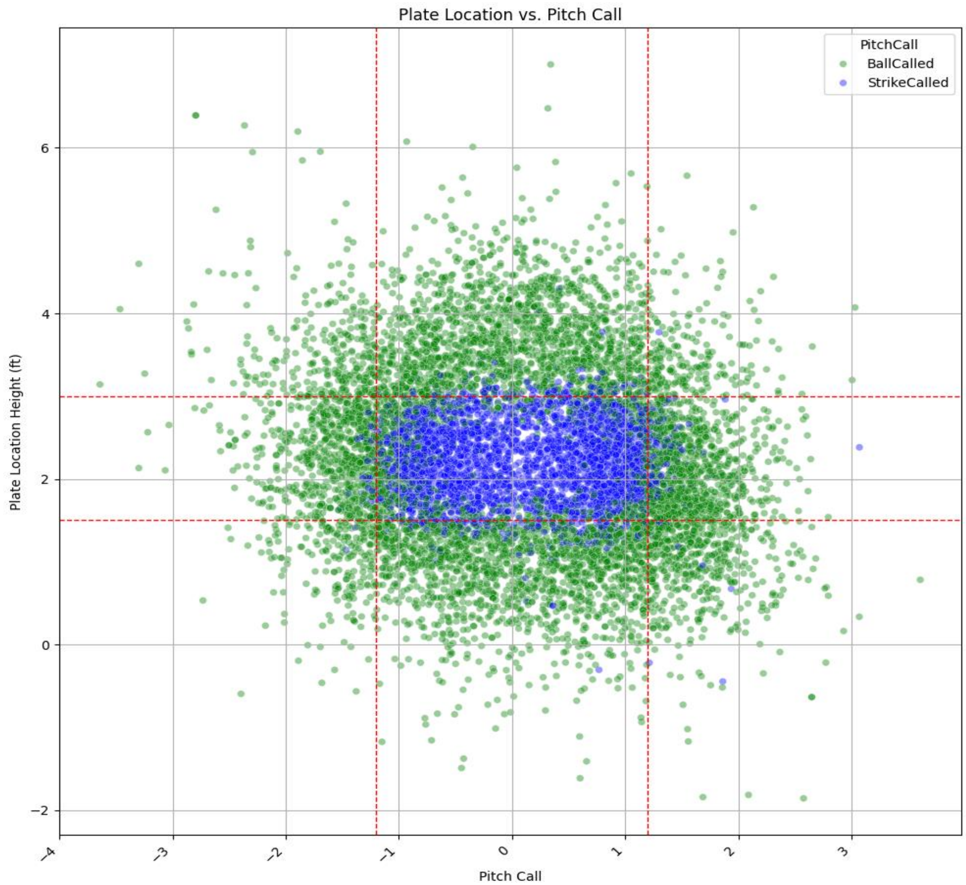
The data set that we used came from the LSU Trackman, which provided a large number of factors, such as type of call, plate height, and many more. A Trackman is a "a premier radar-based launch monitor that delivers precise, real-time data on both club delivery and ball flight, used for golf coaching, fitting, and simulation". It uses dual radar technology to track the ball throughout the entire play. The data we were provided also listed important factors such as who was the pitcher, if the pitch landed in the strike zone, what type of pitch it was, etc. As well as enough data points to provide a clear and full image of what a typical situation would look like during any given game. Since the data was too large at times to run, we also ran tests using smaller sample sets to see if the trends we found consisted throughout all the data. This made it easier to work with and provided more confidence in our findings, when the results came back consistent.

Methodology/ What we have done

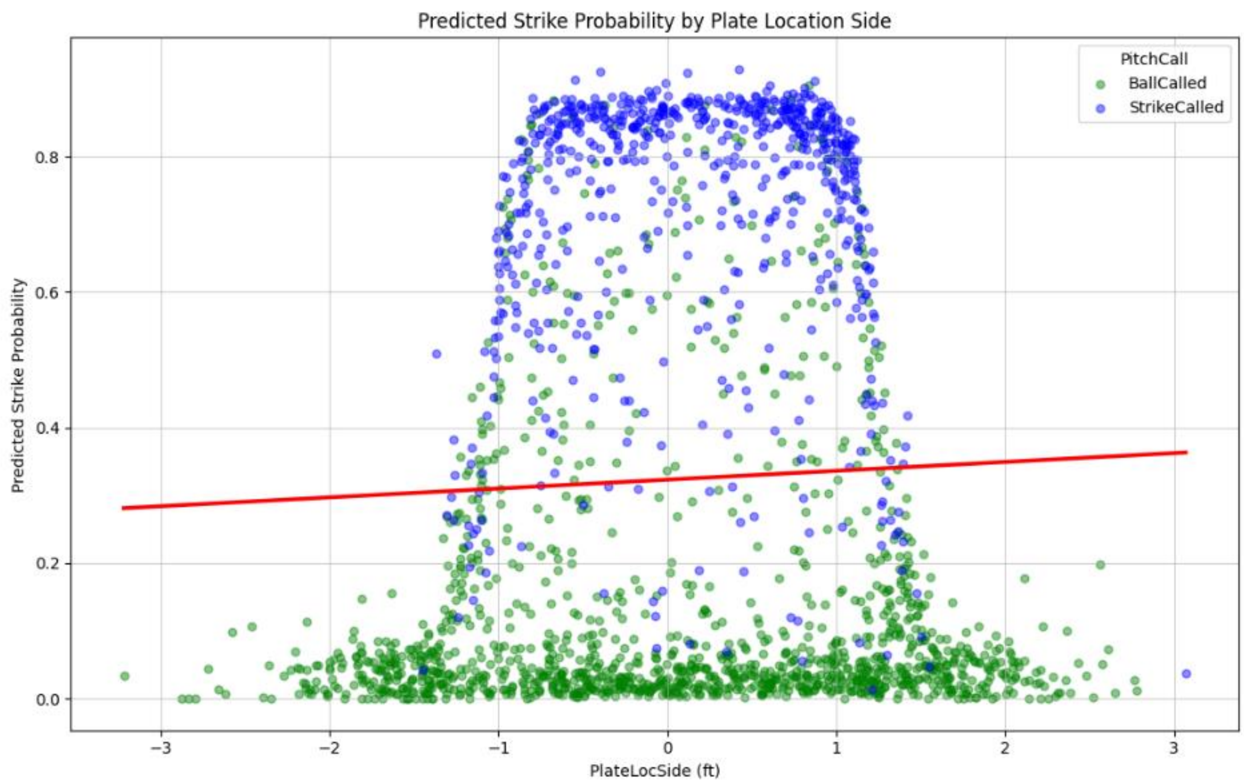
Our method was to build models using Matplotlib and seaborn in Python to generate the model shown here in our rough sketch.



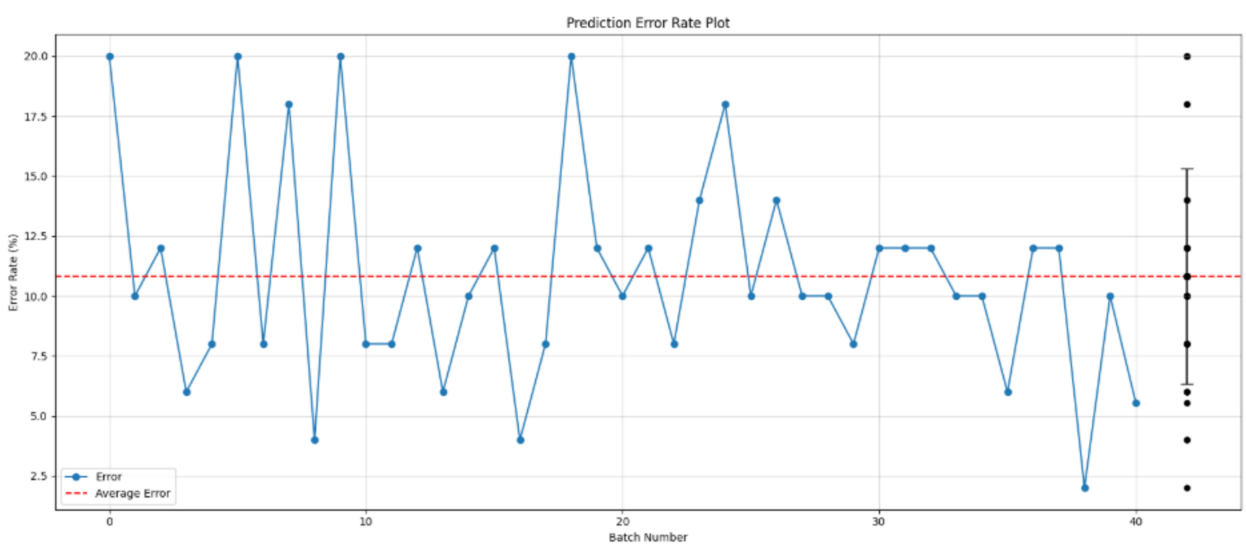
From this picture you can observe that we wanted to start with last year's teams ideas on a strike probability model and build our "catcher frame" around it. We figured to go about this we'd build a new strike zone from the recent LSU Softball Team data and highlight the pitches that can be easily swayed into the strike zone. Firstly, it was necessary for us to create a scatter plot and figure out the true strike zone. Which we found to be 3ft tall and 1.2 ft on either right or left side of the batter plate. Shown here



We also used a random forest model to evaluate catcher framing. It first established the probabilities of the pitch being called a strike or ball based on location, and then we compared these probabilities to the actual pitch outcomes. This allowed us to determine the impact of the catcher framing and observe whether catchers successfully "stole" strikes or "lost" them due to poor framing



The Random Forest model clearly captures the strike zone pattern. Pitches near the center are assigned high strike probabilities, while pitches farther outside receive low probabilities. This indicates the model learned realistic nonlinear decision boundaries.



The average batch error rate is around **11%**, with most batches staying close to this level. This suggests stable and consistent predictive performance across the test data. Random Forest was the strongest model overall, combining high accuracy, high precision, and excellent classification ability.

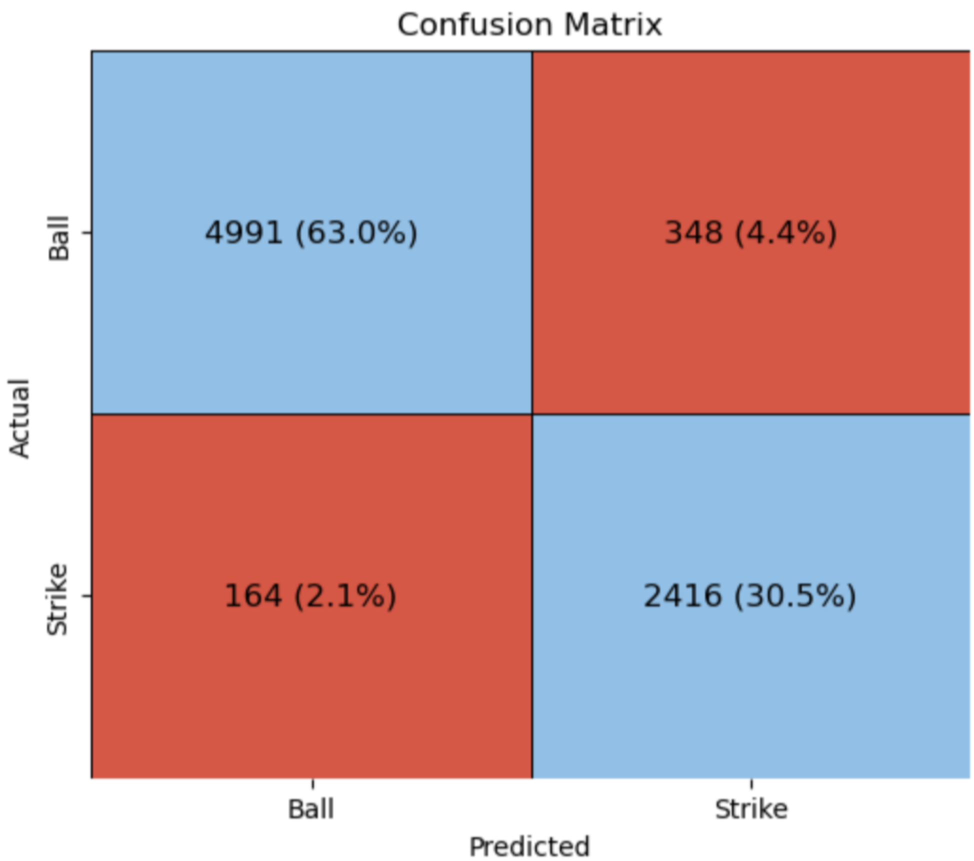
Problems/ Limitations

There weren't too many problems we faced when it came to coding; however, we did struggle trying to determine which type of model would be most accurate and most useful when it came to benefiting the LSU team. To overcome this, we tried to create multiple models that each showed the "shadow zone". It was also hard to determine what to do with the models we discovered. However, after more research we found that the Shadow Zone is a good place for pitchers to aim, because it results in a higher chance of strikes along with making it harder for the batter to hit. With this information, it was clear that the Shadow Zone model we developed could be used to help pitchers know where to aim.

We also had a hard time learning the software. Only one of the members of our group had a lot of experience with coding, so it was a learning curve for the rest of us. We also used multiple programs to code, based on what we were most comfortable with, which made collaborating on different code hard.

Another issue we ran into was that the dataset was much too large to run at once. It would slow down the program, and our computers had a hard time processing all the information. To try and resolve this issue, we had to break the data into different sets, test it, and then compare all the models' side by side. This was more time-consuming and perhaps made our model less accurate, but with the technology we had access to it was the most effective way.

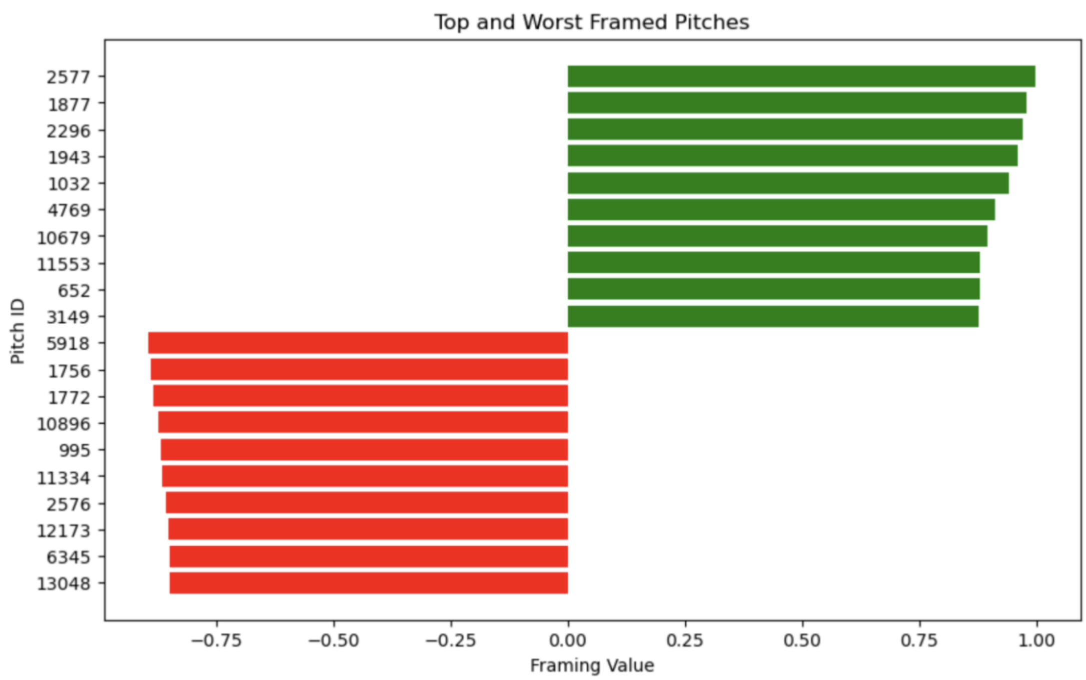
- Model Performance



The random forest model did a great job of learning the strike zone and achieved an overall accuracy of 93.5% on the test set. Based on the confusion matrix, the model correctly classified most pitches, with 4,991 True Negatives (correctly predicted balls, 63.0%) and 2,416 True Positives (correctly predicted strikes, 30.5%). There were 348 False Positives where the model predicted strikes but were actually called balls, and 164 False Negatives that the model predicted as balls but were actually called strikes.

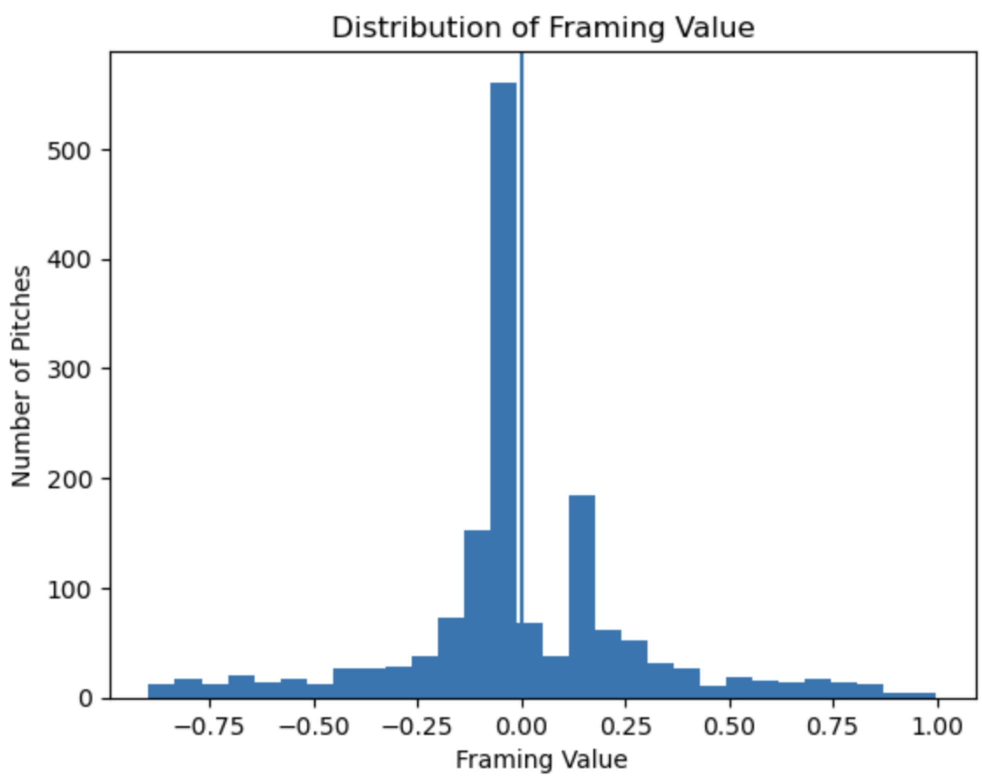
[4]:
```python
# Predict probabilities
probs = model.predict_proba(X_test)[:, 1]

results = X_test.copy()
results['Actual'] = y_test.values
results['ExpectedStrikeProb'] = probs

# Framing value
results['FramingValue'] = results['Actual'] - results['ExpectedStrikeProb']
print(results['FramingValue'])
```

```
8208     0.192500
12803    0.111656
2785     0.113749
3249     0.161969
5045    -0.096864
           ...
2323     0.379298
6653    -0.105265
13325   -0.556106
8600     0.111657
7434    -0.190452
Name: FramingValue, Length: 1584, dtype: float64
```



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.

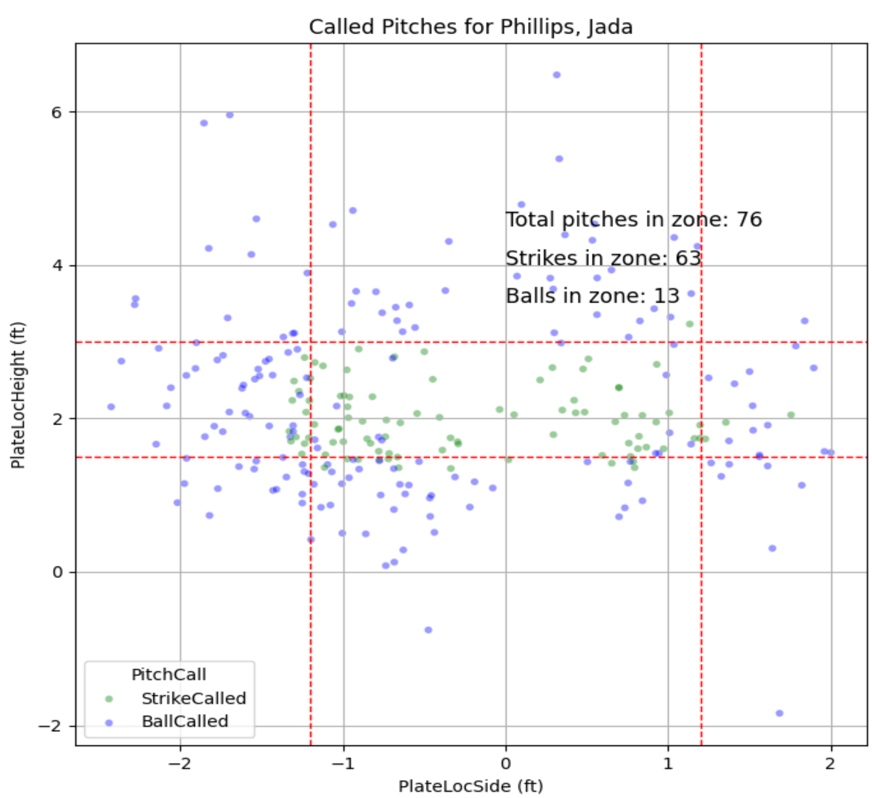
The positive values represent good framing, where the catcher was able to “steal” a strike. It occurs when a pitch is called a strike despite having a low expected probability. For example, a framing value near +0.9 means a pitch only had a 10% chance of being a strike, but the catcher framed it well 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.



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.

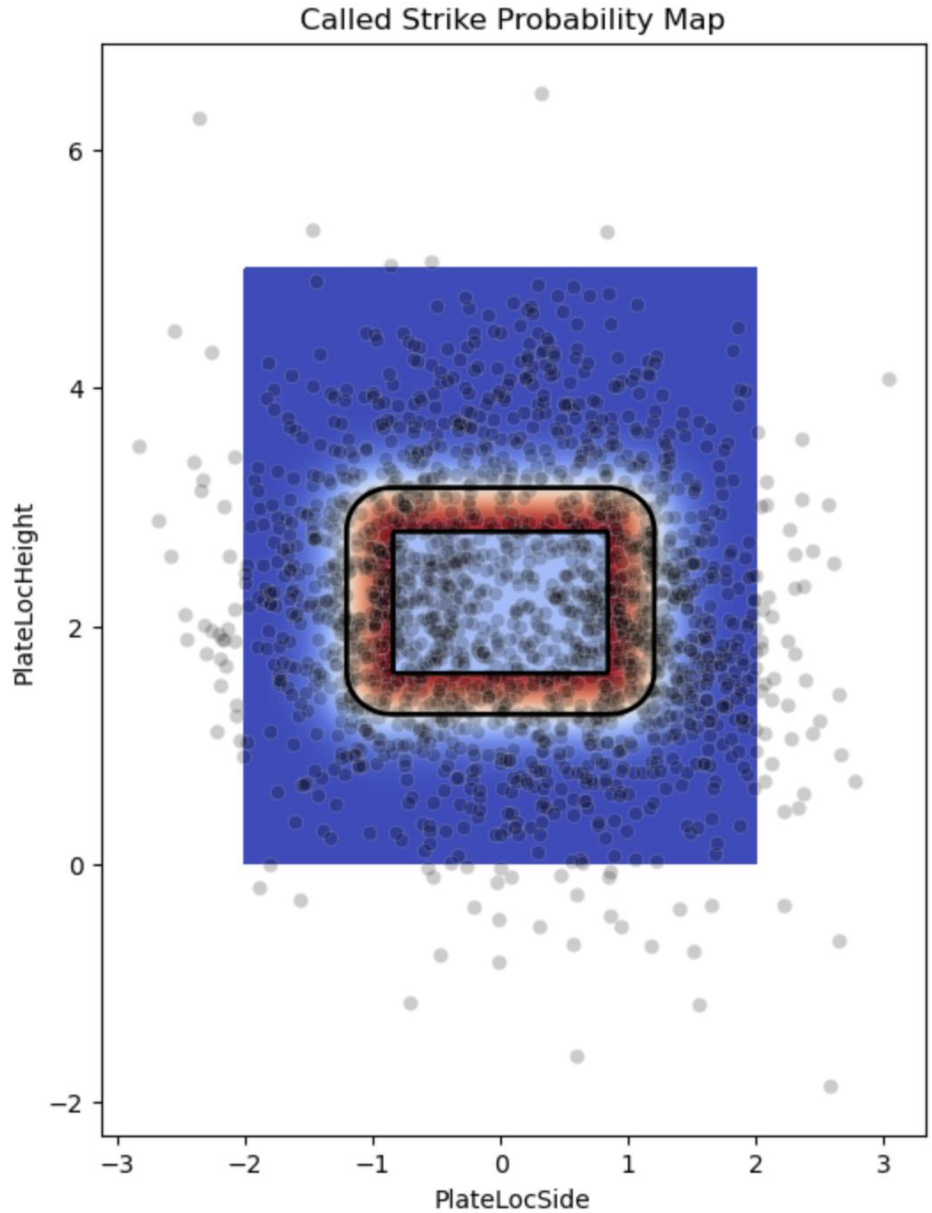
The outliers (tails) more clearly represent distinct catcher performance. Pitches like ID 2577 and 1877 represent exceptional framing, registering values near +1.0, which means that they were highly unlikely strikes that were successfully stolen. On the other hand, pitches like ID 5918 and 1756 represent framing failures, registering values near -0.9, meaning that they were highly probable strikes that were lost.

We were able to model framing for each catcher. For example, here in this model Jada Phillips' strike zone is less than 3ft, but her framing went well as she has strikes called outside on the 1.2ft bounded for the strike zone. This means she was able to favorably move those pitches along the shadow zone and make them a strike.



The Shadow Zone and Heart Zone

The highlight of our project was to model the layers of framing zones. If we could get an estimate of where the shadow zone and heart zone was, the coaches could get an idea of how beneficial their catchers' framing is or if it needs improvement. In the graph, the red zone is the heart zone, and the orange is the shadow zone. But not to overlook it, the white area is the chase zone. It can be observed in the graph that the catcher zones begin at 1ft and end at 1.2 ft which matches our strike zone. It is not a coincidence or inaccuracy that these line up. It means when comparing the two graphs we can see how many balls were caught and moved into the strike zone. Any strikes seen pass the 1ft line in our strike zone graph means they were caught in either the heart, shadow, or chase zone and framed to be a strike.



Conclusion

All in all, the models and computations we produce can help the LSU Softball Pitchers know the best places to throw in accordance with the Shadow Zone. As well as help our LSU Batters know the likelihood of the ball that is out of the Strike Zone is going to be called a strike. This is important because it will help guide their practices in terms of training to become better at hitting balls that are in the Shadow Zone. Also, by pairing a highly accurate Random Forest model with a logical Framing Value calculation, our framework successfully isolates and allows us to clearly see the hidden value of catcher framing. This allows for both macro-level evaluation of a catcher's seasonal performance and micro-level review of specific pitches where strikes were won or lost. For the next steps, we think it would be interesting to investigate ball trajectory after hitting a pitch that was in the Shadow Zone. This could be beneficial to seeing and training Batters to better hit these balls.

Work Cited

“Catcher Framing Evaluation and Reporting with 6-4-3 Charts – Softball Edition.” *6-4-3 Charts*, https://643charts.com/catcher-framing-evaluation-and-reporting-with-6-4-3-charts-softball-edition/

Franke, Kai. “Strike Probability Model and Catcher Framing Using Random Forest.” *Medium*, 24 May 2022, https://medium.com/@kaifranke3/strike-probability-model-and-catcher-framing-using-random-forest-7b482d6f159f

“Loss Functions.” *PyTorch Documentation*, PyTorch nn Module Loss Functions

Nestico, Thomas. “Modelling Batter Decision Value.” *Medium*, 1 Dec. 2023, https://medium.com/@thomasjamesnestico/modelling-batter-decision-value-dac74c55e20a

Nestico, Thomas. “Classifying MLB Pitch Zones and Predicting MiLB Zones.” *Medium*, 17 Nov. 2023, https://medium.com/@thomasjamesnestico/classifying-mlb-pitch-zones-and-predicting-milb-zones-7e95cf308254

“Radar | Glossary of Terms.” *TrackMan Baseball Support*, Radar Glossary Of Terms