How Pitch Count Influences Fringe Pitch Calls

Kanyen Chan

UCLA Department of Atmospheric and Oceanic Sciences

December 2025

1 My Project

I applied machine learning techniques to investigate how pitch count influences an umpire's decision on (borderline) fringe pitches. Below is my report.

2 Introduction

The purpose of this report is to ascertain whether baseball fans believe that home-plate umpires are more likely to rule fringe pitches differently depending on the pitch count. Precisely, we aim to evaluate two scenarios by controlling one of the count's components:

- **Scenario 1 – Hitter's count:** 3 balls and {0, 1} strikes.
- **Scenario 2 – Pitcher's count:** 2 strikes and {0, 1, 2} balls.

Consider the diagram below:

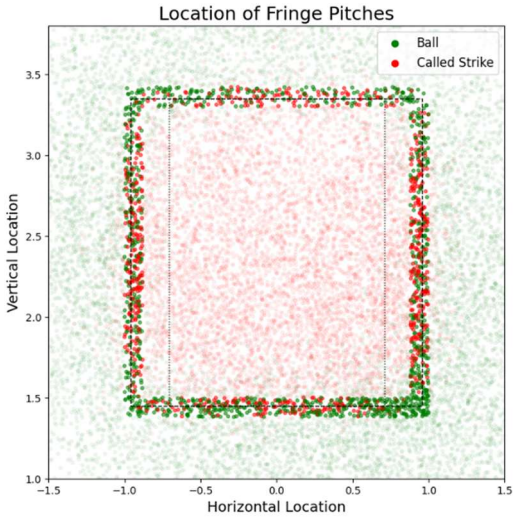

Figure 1: Scatterplot of Pitches

Definitions:

- **The Universal Strike Zone:** Defined by the inner rectangle, the strike zone has a horizontal boundary measuring 17 inches in width [1]. The vertical boundary remains subjective, typically determined by the space between the batter's shoulders and the top of the uniform pants [2]. For our data, we defined the vertical length by examining the data to see where the majority of overlap between balls and strikes occurred.
- **Fringe-Pitch Zone (Buffer Zone):** Defined by the outer rectangle, representing the area with the most variance in a pitch being called a ball versus strike. For this study, the buffer zone extends 1.44 inches (0.12 feet) beyond the true strike zone

Note: The diameter of a baseball ranges from 2.86 to 2.94 inches [3]. A pitch is considered a strike if any part of the ball crosses into the strike zone.

As previously mentioned, the goal of this study is to determine how pitch count influences an umpire's decision to call a pitch a ball or a strike. With advances in technology, teams are increasingly focused on gaining a competitive edge. By understanding how umpires treat different scenarios, players can make more informed decisions about which pitches to throw, their locations, and which to anticipate as hitters.
Also, in the era of sports gambling, such statistical insights can also inform smarter betting strategies, though the smartest choice is to always avoid gambling altogether.

To address this problem, we use the dataset provided through GitHub and apply a supervised machine learning approach, as the data includes labeled outcomes. Because the objective is to predict whether an umpire calls a pitch a ball or a strike, a classification method is appropriate. Accordingly, we will employ logistic regression and random forests to model umpire decisions on fringe pitches. We will visualize our data using box plots, ROC curves, confusion matrices, feature importance, and histograms. Our analysis indicates that pitch count does indeed influence an umpire's ruling though the location of the pitch has the greatest influence.

Data

Below is an overview of the dataset.

	Unnamed: 0	pitch_type	release_speed	pitcher	description	balls	strikes	plate_x	plate_z	home_score	away_score
0	4	SL	80.9	572888	ball	0	1	1.4999	1.6595	7	9
1	5	FT	94.5	572888	called_strike	0	0	0.4118	1.7614	7	9
2	9	FT	94.6	572888	called_strike	3	1	0.0122	2.5461	7	9
3	10	FT	94.6	572888	ball	2	1	-0.8732	3.5043	7	9
4	11	SL	81.6	572888	ball	1	1	-1.9930	2.3081	7	9
...	...	...	...	...	...	...	...	...	...	...	...
20523	39989	FF	93.5	592426	ball	1	0	0.5788	1.4362	2	0
20524	39990	SI	91.8	453172	called_strike	0	0	0.3628	1.9900	5	5
20525	39993	FF	90.7	598287	ball	0	0	-0.2451	1.4081	1	5
20526	39994	SI	91.2	446372	called_strike	0	2	0.8371	2.9895	2	0
20527	39995	SL	84.2	596816	called_strike	1	0	-0.9636	2.5847	0	2

20528 rows × 11 columns

Figure 2: Data

Within the dataset, there are eleven features.

1. Unnamed: 0
    - This column adds no intrinsic value to our dataset; thus, we will omit this column
2. pitch_type
    - This is the type of pitch the pitchers throw
        - SL represents slider
        - FT represents two-seam fastball
        - FF represents four-seam fastball

3. release_speed
    - This is the miles per hour of a designated pitch
4. pitcher
    - This is the pitcher ID, this will not be relevant for our model
5. description
    - Our target variable; describes if the umpire called a pitch a ball or strike
6. balls
    - Given the pitch count, how many balls has the pitcher thrown
7. strikes
    - Given the pitch count, how many strikes has the pitcher thrown
8. plate_x
    - Horizontal location of pitch
9. plate_z
    - Vertical location of pitch
10. home_score
    - The score of home team
11. away_score
    - The score of away team

Preprocessing Steps:

The first involves loading the dataset into a Google Colab notebook. We used the pandas library to load the csv file as a dataframe object.

The goal is to only look at pitches inside the buffer zone, thus our dataframe below does as such.

```
#Step 1: Define Strike-Zone and Fringe Parameters
x_min, x_max = -0.88,0.88
z_min, z_max = 1.5, 3.3
buffer = 0.12 #adds extra 0.12 ft to the strike zone ("fringe area")

#Step 2: Obtain the fringe pitches
#Gets pitches that are inside the extended (buffered) strike zone but outside the true strike zone
df_fringe = df[
    (
      (df['plate_x'] >= x_min - buffer) &
      (df['plate_x'] <= x_max + buffer) &
      (df['plate_z'] >= z_min - buffer) &
      (df['plate_z'] <= z_max + buffer)
    ) &
    ~(
        (df['plate_x'] >= x_min) &
        (df['plate_x'] <= x_max) &
        (df['plate_z'] >= z_min) &
        (df['plate_z'] <= z_max)
    )
].copy()
```

Figure 3: Preprocessing Step (1)

After, I made a copy of the original data frame.

```
#Get the train and test data from the dataframe
df_clean = df_fringe.copy()

#Convert "ball" and "called_strike" to numeric values. 1 for 'called_strike', 0 for ball
df_clean.description = [1 if value == 'called_strike' else 0 for value  in df_fringe.description]

#Define input features and target variable
X = df_clean[['release_speed','balls','strikes','plate_x','plate_z','home_score','away_score']]
y = df_clean['description']
```

Figure 3: Preprocessing Step (2)

We first converted the 'description' column to {1 for strike, 0 for ball}
Then we defined the input features in line 11. We decided to omit: Unnamed: 0, pitch_type, pitcher.
We define the target variable, y, as 'description'.
Finally, train_test_split from SKLEARN.MODEL_SELECTION was used to the data into a training and testing set. 80% of the data was used to train the models while 20% for the test.

3 Modeling

For this project, we will deploy two supervised classification methods.

1. Logistic Regression
2. Random Forests

Then we will utilize these models to make predictions based off of hypothetical pitch counts
As mentioned in the introduction, (1) Hitter's Count & (2) Pitcher's Count

3.1 Logistic Regression

The goal of Logistic Regression for this specific dataset is to determine whether a pitch in the buffer zone will be called a ball or a strike by the umpire. However, a pitfall of logistic regression is that it assumes linearity between the input features and the target variable. After the preprocessing steps seen above, I then applied my logistic regression model with 5-fold cross-validation using Scikit-learn LogisticRegressionCV. Cross-validation splits the training data into five subsets, trains the model on four of them, and validates it on the remaining one. This process is repeated across all folds, and the model automatically selects the best regularization hyperparameter. In addition, regularization helps reduce overfitting by penalizing large coefficients.
The model resulted in an accuracy of 54.45% and a log-loss value of 0.687.

3.2 Random Forests

A supervised Random Forest classification model is an ensemble learning method which utilizes a collection of decision tree classifiers (denoted by the parameter: n_estimators) each trained on different random subsets of the data. The final prediction is made by averaging the outputs of these individual trees. This ensemble approach improves predictive accuracy and aids in

reducing overfitting, since combing a diverse set of trees is more robust than a single decision tree.
In our model, each decision tree was restricted to a maximum depth of 5, and the forest consisted of 200 decision trees.
The model resulted in an accuracy of 66.67% and a log-loss value of 0.638.

3.3 Hypothetical Tests

After training our models, we proceed to deploy them to perform the hypothetical tests. Without loss of generality, we focus on the case of logistic regression, as the evaluation procedure is identical for other models.

```
#----PERFORMING HYPOTHETICAL TESTS (LOGISTIC REGRESSION)------#
import numpy as np

test_3_balls = df_fringe.copy()
test_2_strikes = df_fringe.copy()

#Hitter's count -> Testing the situation when it's advantage hitter
#1 Sets the balls column to 3
#2 Randomizes the strike column to either 0 strikes or 1 strike
test_3_balls['balls'] = 3 #Sets the balls column to 3
test_3_balls['strikes'] = np.random.choice([0,1], size=len(df_fringe), replace=True)

#Pitcher's count -> Testing the situation when it's advantage pitcher
#1 Sets the strike column to 2 strikes
#2 Randomizes the ball column to either 0, 1, 2 balls
test_2_strikes['strikes'] = 2
test_2_strikes['balls'] = np.random.choice([0,1,2], size=len(df_fringe),replace=True)
```

Figure 4: Hypothetical Tests (1)

First, we create two copies of the fringe-pitch dataframe to represent our hypothetical scenarios: one in which the pitch count is 3 balls (a hitter's count) and another in which the count 2 strikes (a pitcher's count).
For hitter's count scenario, we set the 'balls' column to 3 and assign the 'strikes' column a random value of either 0 or 1 strikes.
For pitcher's count scenario, we set the 'strikes' column to 2 and assign the 'balls' column a random value of 0, 1, or 2 balls.

```
#Make Predictions (probablities) on 3-balls and 2-strikes
features = [
    'release_speed','balls','strikes',
    'plate_x','plate_z','home_score','away_score'
]

prob_3_balls =  lr_model.predict_proba(test_3_balls[features])[:,1]
prob_2_strikes = lr_model.predict_proba(test_2_strikes[features])[:,1]

mean_3_balls = np.mean(prob_3_balls)
mean_2_strikes = np.mean(prob_2_strikes)
```

Figure 5: Hypothetical Test (2)

Finally, we generate predicted probabilities for both hypothetical scenarios and compute the mean predicted probability (i.e. the expected value) for the 3-balls case and the 2-strikes case. These averages represent the model's estimated likelihood that the umpire will call a strike under each count scenario.

4 Results

Accuracy of Logistic Regression: 54.45%
Log-loss value of Logistic Regression: 0.6865
Accuracy of Random Forest: 66.67%
Log-loss value of Random Forest: 0.6379

Plotted below are the two confusion matrices for each of our models:

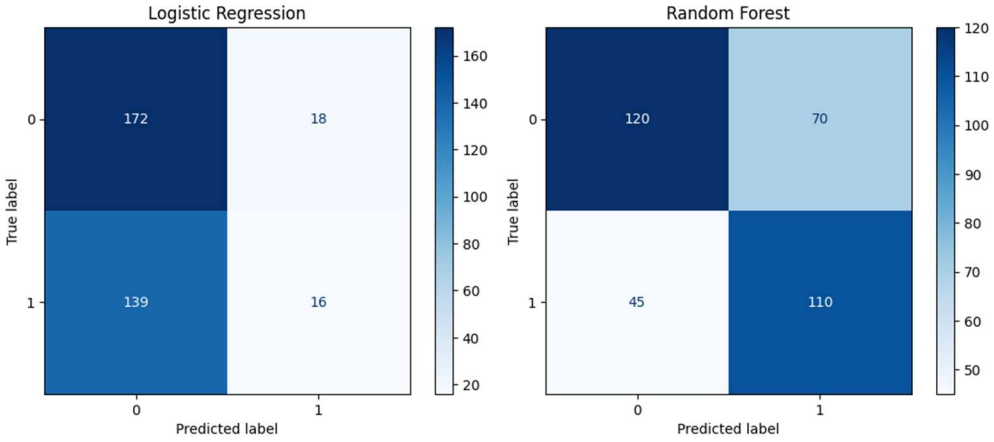

Figure 6: (a) Logistic Regression (b) Random Forest

Plotted below are the two ROC curves for each of our models:

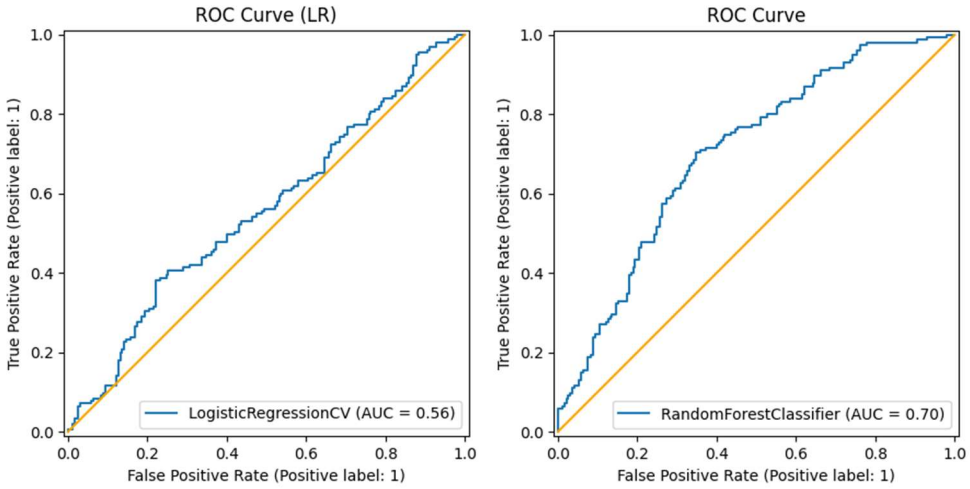

Figure 6: (a) Logistic Regression (b) Random Forest

Plotted below are the box plots of our hypothetical tests:

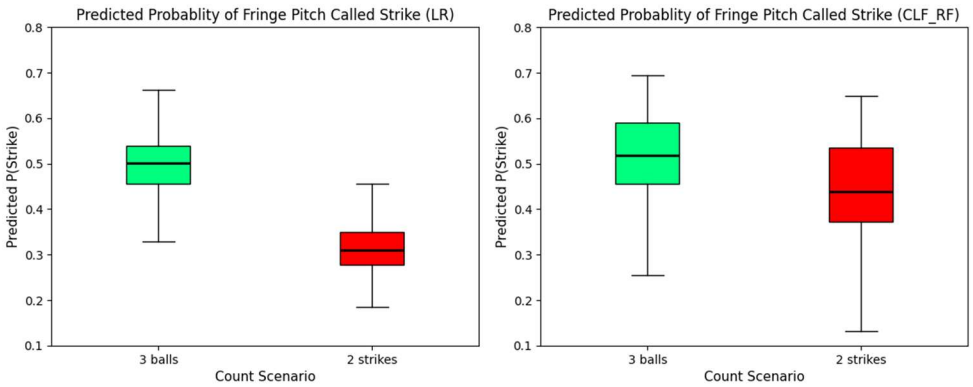

Figure 7: (a) Logistic Regression (b) Random Forest

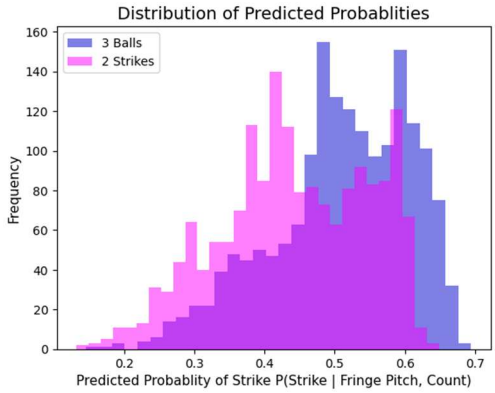

Figure 8: Histogram

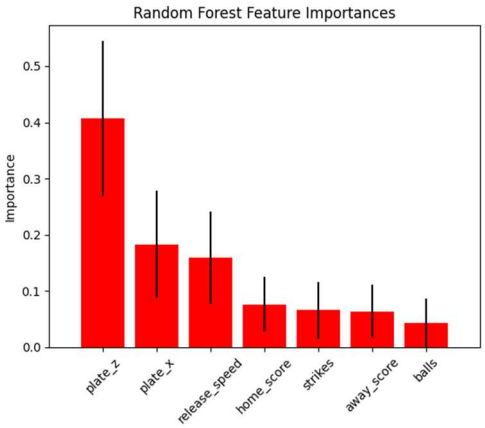

Figure 9: Feature Importance

5 Discussion

The accuracies of our model were relatively mediocre, with accuracies of 54.45% and 66.67% for logistic regression and random forest, respectively. These percentages reflect each model's success rate in predicting when an umpire would call a pitch a strike. The improvement in accuracy of the random forest suggests that the relationships among the variables are partly non-linear, which the logistic regression was unable to capture.

We then calculated the log-loss values as our error metric. In essence, log-loss penalizes incorrect predictions more substantially when the model is confident, thus a lower value indicates stronger probability estimates. The logistic regression and random forest models produced log-loss values of 0.6865 and 0.6379, respectively. This suggests neither model can cleanly distinguish a barrier which separates balls from strikes.

The confusion matrices help us identify how the two models operate in predicting how an umpire calls a fringe pitch. For Logistic Regression, there were 139 false negatives, indicating the model frequently predicted a ball, when in reality, was a strike. This indicates our model was super conservative, heavily labeling fringe pitches as balls.
Our Random Forest model did significantly better. The model consisted of 110 true positives and 120 true negatives, indicating the model correctly predicted balls and strikes. This suggests the model was far more balanced predictor and was able to capture non-linear relationships the logistic regression struggled to model.

Figure 6 compares the ROC curves for Logistic Regression and Random Forest models. The area under curve (AUC) of the logistic regression model was 0.56, which indicates that the model was only slightly better than taking a random guess if a pitch was going to be a ball or strike, which is represented by the orange line with an AUC of 0.5.
In contrast, our Random Forest model obtained an AUC of 0.70, indicating the model was able to better distinguish between the two classes.

In Figure 7, the box plots yield the following results.

Logistic Regression model:

- Mean probability for hitter's count: 49.99%
- Mean probability for pitcher's count: 31.39%

Random Forest model:

- Mean probability for hitter's count: 50.99%
- Mean probability for pitcher's count: 44.25%

From both of our models, obverse that an umpire's likelihood of calling a strike is modestly influenced by the pitch count. In particular, umpires appear more inclined to call a strike when

the count favors the hitter, which may suggest game context subtly influences decision-making in the buffer zone.

Figure 8 represents two overlapping histograms, each showing the predicted probability a pitch is called a strike conditioned it is a fringe pitch and given the count. Backing up our box plot data, this figure illustrates umpires are more conservative when the pitcher is ahead in the count, and more generous when the hitter is ahead. Thus, the pitch count remains a subtle yet influential factor on how the umpire calls the game.

Figure 9 sheds light on feature importance for our random forest classifier. It reveals that while game context subtly influences umpire decision-making, the biggest influence, with approximately 40% of the model's predictive power, is the vertical location of the pitch. The horizontal location and release speed emerged as the second and third most important features, respectively. These results reinforce that the pitch's location in the buffer zone remains the primary source of an umpire's call.

One notable limitation of our model is the absence of pitch type as an input variable. Since release speed ranked third in importance, this result suggests slower pitches (e.g., curveballs) and faster pitches (e.g., fastballs) may influence an umpire's decision more strongly than pitch count itself.

6 Conclusion

In summary, this project aimed to evaluate how pitch count affects an umpire's likelihood of calling a borderline pitch a ball or a strike. Using two different machine learning models, we examined the relationship between count context and called-strike probability. Our findings were successful as they indicate that umpires are more inclined to award a fringe pitch as a strike in hitter's counts compared to pitcher's counts, suggesting that the games situation subtly influences strike-zone judgment.

However, our study is not without limitations. Several potentially important features, such as 'pitch_type' and 'pitcher' were excluded from the analysis. The omission of pitch type is significant because the movement profile of a curveball differs substantially from that of a fastball, and these visual differences can influence an umpire's perception. Likewise, individual pitchers release the ball from varying arm slots and angles, which can affect how the pitch appears as it crosses the plate. Excluding these two variables likely reduced the models' ability to capture some of the real-world complexity inherent in umpire decision-making.

Moreover, our study revealed more than we sought after in the beginning. The feature importance analysis from our random forest classifier revealed that vertical location, followed by horizontal location, played the most significant role in predicting an umpire's call. Even after removing the features such as 'pitch_type' and 'pitcher', the model consistently relied on where the pitch crossed the plate to make its predictions. This suggests that, despite the circumstantial

effects introduced by the pitch count, the physical location of the pitch continues to command an umpire's decision-making process. In other words, pitch count can marginally influence the likelihood of a called strike, but the fundamental definition of the strike zone remains the primary source of these calls.

To further extend this work in a future project, as mentioned, we can incorporate input features such as 'pitch_type' and 'pitcher', both of which play important roles in how a pitch is perceived over the plate. Additionally, we can implement more advanced modeling approaches, such as neural networks and/or ensemble methods. This can help us better capture subtle patterns and nonlinear relationships in the data. Another interesting idea would be to capture data on individual umpires, as each umpire exhibits their own distinct tendencies on how they call a game. Similarly, we can do the same for specific pitchers, since differences in release angle, arm slot, and pitch profile may influence the umpire's decision at the plate. Moreover, we can implement situational context to our data. These include but are not limited to number of runners on base, if there are runners in scoring position (second base or third base), the score differential between teams, and the inning. This could possibly reveal how external pressures can influence an umpire's decision. Finally, increasing the size of the dataset across multiple seasons could allow for better identification in trends, and potentially improve the predictive power of future models.

7 References

[1] Boyle, W. (2018, June 26). *Prospectus feature: The universal strike zone*. Baseball Prospectus. https://www.baseballprospectus.com/news/article/40891/prospectus-feature-the-universal-strike-zone/

[2] *Strike zone: Glossary*. MLB.com. (n.d.). https://www.mlb.com/glossary/rules/strike-zone

[3] Daisey. (2020). *Baseball Dimensions & Drawings*. RSS. https://www.dimensions.com/element/baseball