DSCI 631: 001 APPLIED MACHINE LEARNING FOR DATA SCIENCE

FINAL PROJECT

Due: June 9th, 2024

Enhancing Baseball Strategy

Classifying Pitch Outcomes on Taken Pitches

Caleb Miller
Hashim Afzal
Robert Logovinsky

Table of Contents

Data and Task:

The goal of this project is to develop a classification model that will predict whether a pitch will be called a strike (1) or a ball (0) when the batter does not swing at it. Predicting whether a taken pitch will be called a strike is a key project for baseball teams. This model allows teams to leverage predictions of strike or ball calls for various applications, such as evaluating catcher framing, making swing decisions, and analyzing umpire tendencies.

The dataset that we acquired is a collection of pitches from the 2022 MLB Season where the batter did not swing at the pitch. This dataset was sourced from Baseball Savant and has 351,062 rows and 20 columns.

Column Breakdown:

- game_pk : unique identifier for a specific game - data type is an integer
- game_date : the date on which the game occurred in the format MM/DD/YYYY - data type is date
- at_bat_number : unique identifier for a specific plate appearance - data type is an integer
- pitch_number : pitch number within the plate appearance - data type is integer
- pitch_type : identifies what type of pitch was thrown - data type is string
    - can be one of:
        - CH : changeup
        - CS, CU : curveball
        - EP : eephus
        - FA, FF : four seam fastball
        - FC : cutter
        - FS : splitter
        - KC : knuckle curve
        - KN : knuckleball
        - SI : sinker
        - SL : slider
- pitcher_name : name of the pitcher in lastname, firstname format - data type is string
- pitcher : unique identifier for pitcher - data type is integer
- batter : unique identifier for batter - date type is integer
- catcher : unique identifier for catcher - data type is integer
- description : describes whether a pitch was called a ball or strike - data type is string
- zone : describes the zone location a ball when it crosses the plate - data type is integer
- stand : whether the batter is left-handed or right-handed - data type is string
    - can be one of:
        - L : left-handed
        - R : right-handed
- p_throws : whether the pitcher is left-handed or right-handed - data type is string
    - can be one of:

        - L : left-handed
        - R : right-handed
- balls : how many balls are in the count at the time of the pitch - data type is integer
- strikes : how many strikes are in the count at the time of the pitch - data type is integer
- plate_x : horizontal position of the ball when it crosses the plate - data type is float
    - center of the plate is 0,0, units in feet
- plate_z : vertical position of the ball when it crosses the plate - data type is float
    - the ground is 0,0, units in feet
- sz_top : top of the batter's strike zone - set by the operator when the ball is halfway to the plate - data type is float
- sz_bottom : bottom of the batter's strike zone - set by the operator when the ball is halfway to the plate - data type is float
- broadcast : Link to a video of the pitch - data type is string

Evaluation Metrics:

Since we are working on a classification problem, our metrics for evaluation include accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted instances to the total instances. Precision evaluates the proportion of true positive predictions among all positive predictions made by the model, indicating its ability to avoid false positives. Recall assesses the model's capability to identify all actual positive instances by measuring the proportion of true positive predictions out of all actual positives, thus highlighting its ability to avoid false negatives. The F1 score provides a combination of precision and recall, offering a balanced measure when there is an uneven class distribution.

Preprocessing:

Null Values

Column	Quantity
pitch_type	177
zone	195
plate_x	195
plate_z	195
sz_top	195
sz_bot	195
broadcast	3

We had 195 rows with null values in them. This represents a negligible portion of the data (0.0005%). Since this represents such a small portion of our data, the rows with null values were dropped.

Dropping Columns

We dropped 'pitcher_name' because its data is captured in 'pitcher_id' (this is another indicator of who the pitcher is).

Data Manipulation

We changed the format of the 'description' column from 'ball' and 'called_strike' to 0 and 1 and renamed it to 'is_strike'. Additionally, we changed the format of the 'stand' and 'p_throws' columns from 'R' and 'L' to 0 and 1 and changed the names to 'pitches_lefty' and 'bats_lefty'. Lastly, we updated the datatype of 'zone' from float to integer (this is because we were going to use this as a categorical feature - the decimal precision was unnecessary).

Exploratory Data Analysis (EDA):

T-Tests

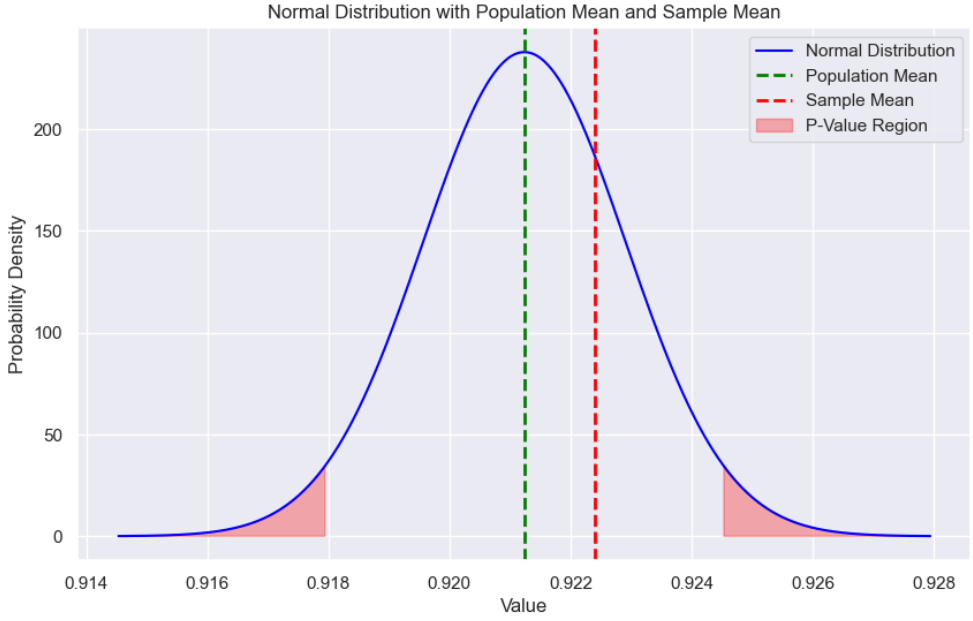
All Combinations of Pitcher and Batter Orientation

We began by examining various combinations of pitcher and batter-handedness to determine if this would influence an umpire's ability to accurately call strikes and balls. The strike zone and viewing angles differ depending on whether the pitcher or batter is left or right hand dominant. Given that most pitchers and batters are right-handed, we hypothesized that there might be a discrepancy in the accuracy of calls based on the different handedness orientations of pitchers and batters.

To determine whether or not there is an association between pitcher and batter orientation, t-tests are crucial.

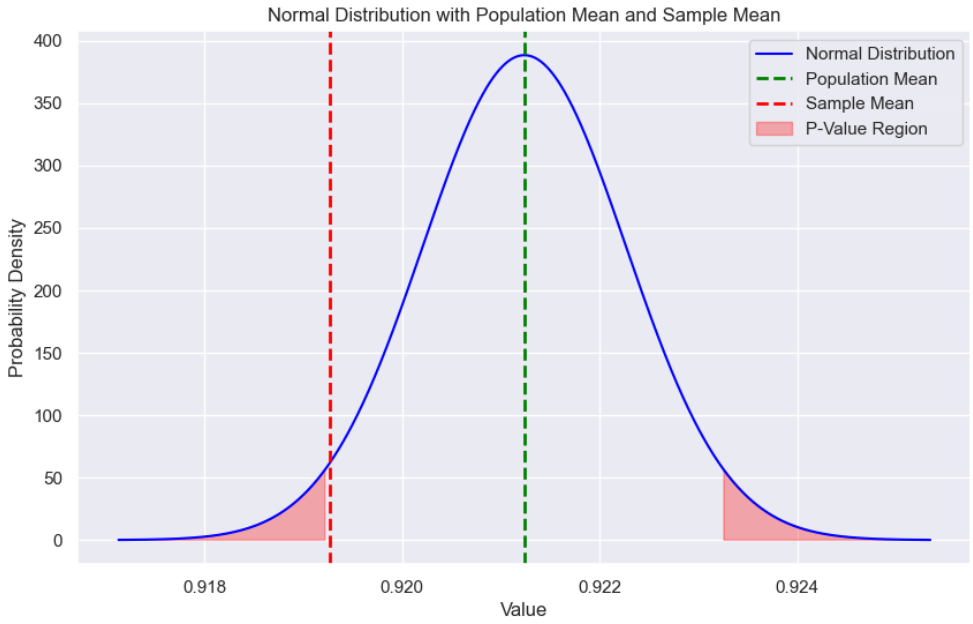
For all combinations of pitcher and batter handedness, we failed to reject our null hypothesis that there is no association between the handedness of the pitcher/batter and the correctness of the umpire's call.

Left-Handed Pitchers to Left-Handed Batters



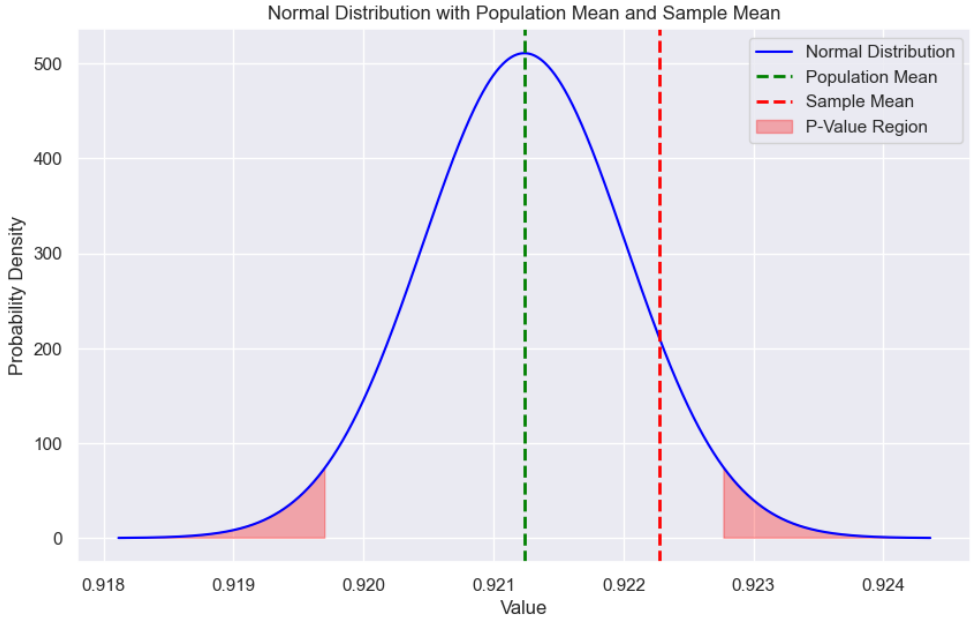
The sample mean does not deviate from our population mean at our alpha of 0.05. Thus, we fail to reject H0. Therefore, we fail to find an association between correct call percentage and left-handed pitchers to left-handed batters.

Left-Handed Pitchers to Right-Handed Batters



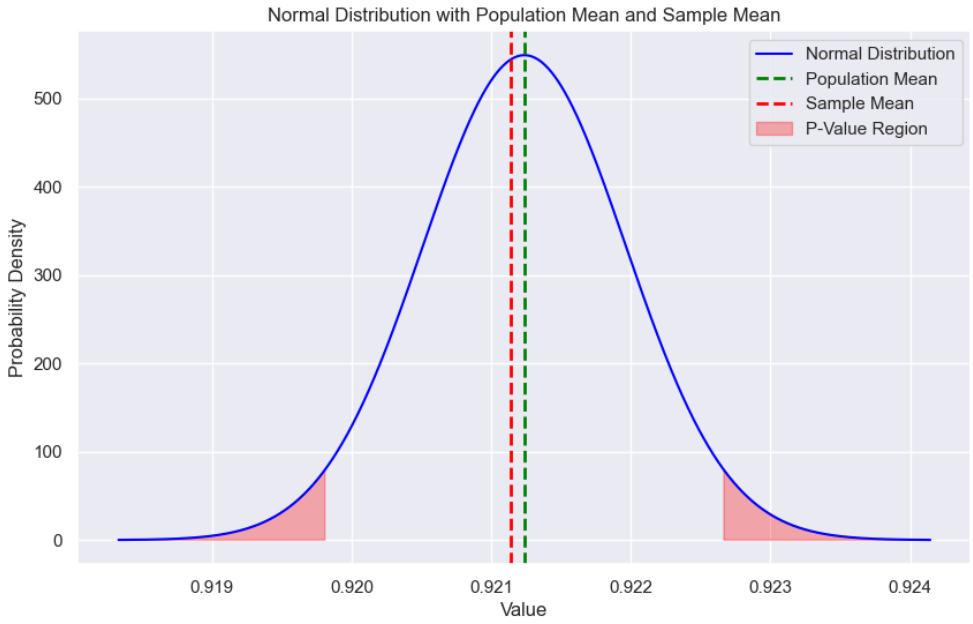
The sample mean does not deviate from our population mean at our alpha of 0.05. Thus, we fail to reject H0. Therefore, we fail to find an association between correct call percentage and left-handed pitchers to right-handed batters.

Right-Handed Pitchers to Left-Handed Batters



The sample mean does not deviate from our population mean at our alpha of 0.05. Thus, we fail to reject H0. Therefore, we fail to find an association between correct call percentage and right-handed pitchers to left-handed batters.

Right-Handed Pitchers to Right-Handed Batters



The sample mean does not deviate from our population mean at our alpha of 0.05. Thus, we fail to reject H0. Therefore, we fail to find an association between correct call percentage and right-handed pitchers to right-handed batters.

Pitch Type

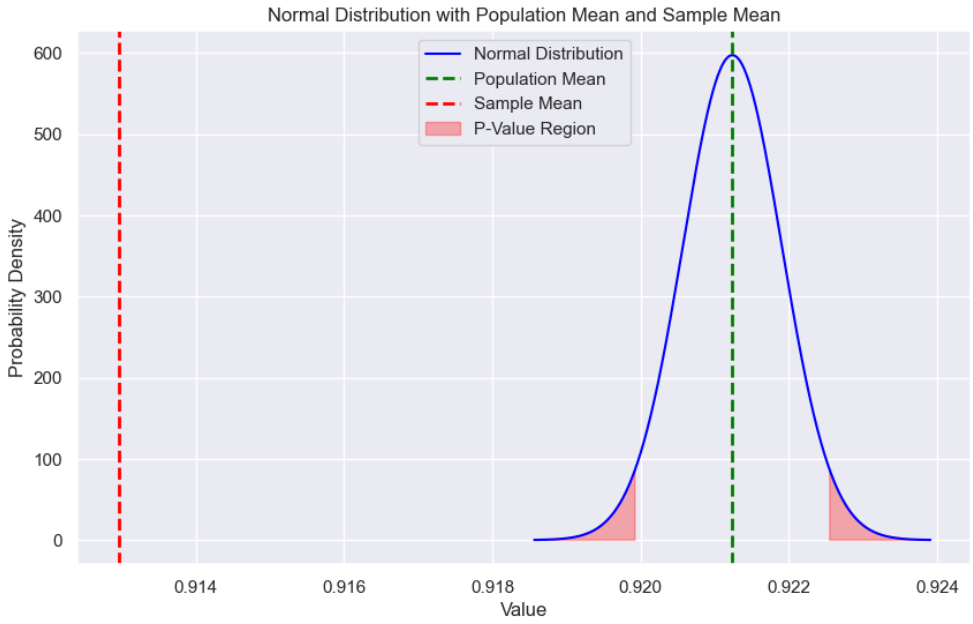
Next, we examined various pitch types, including curveballs, knuckleballs, sliders, and fastballs, etc. to determine if the type of pitch affects an umpire's ability to accurately call strikes and balls. Each pitch type has a different trajectory, and its movement can alter significantly as it traverses the space from the pitcher to the plate, which may have an impact on the umpire's judgment.

To reduce the amount of t-tests, we bucketed pitch types into three categories:

1. Fastballs - Four-Seam Fastball, Sinker
2. Breaking Balls - Slider, Knuckle-Curve, Curveball, Knuckle-Curve
3. Offspeed - Changeup, Splitter, Knuckleball, Eephus

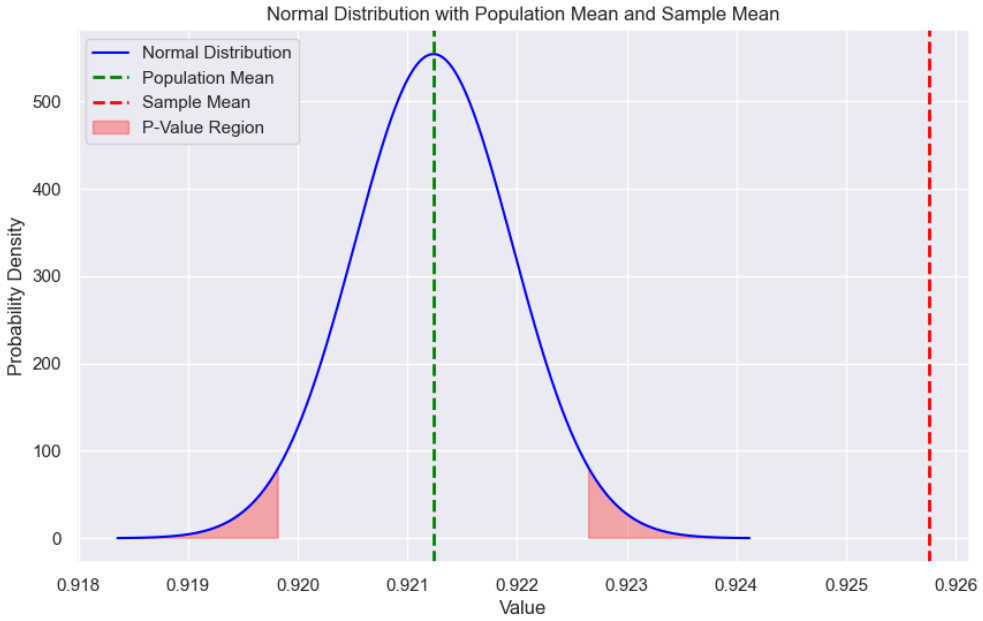
After analysis, we reject the null hypothesis for all three pitch groups and find that the type of pitch *is* statistically significant in determining correct call percentage.

Fastballs



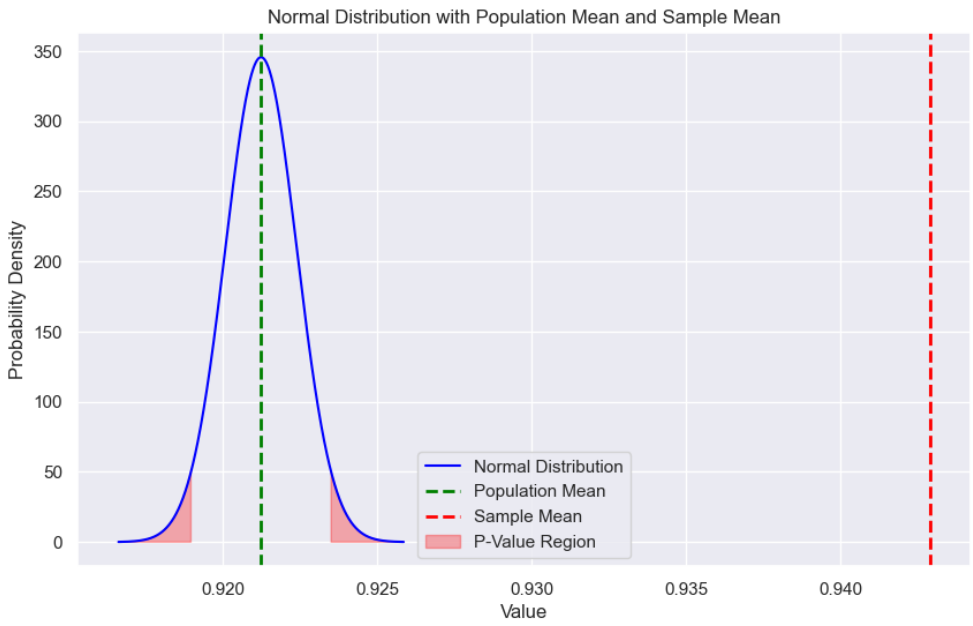
The sample mean deviates from our population mean at our alpha of 0.05. Thus, we reject H0 and accept HA. Therefore, we found an association between correct call percentage and fastballs.

Breaking Balls



The sample mean deviates from our population mean at our alpha of 0.05. Thus, we reject H0 and accept HA. Therefore, we found an association between correct call percentage and breaking balls.

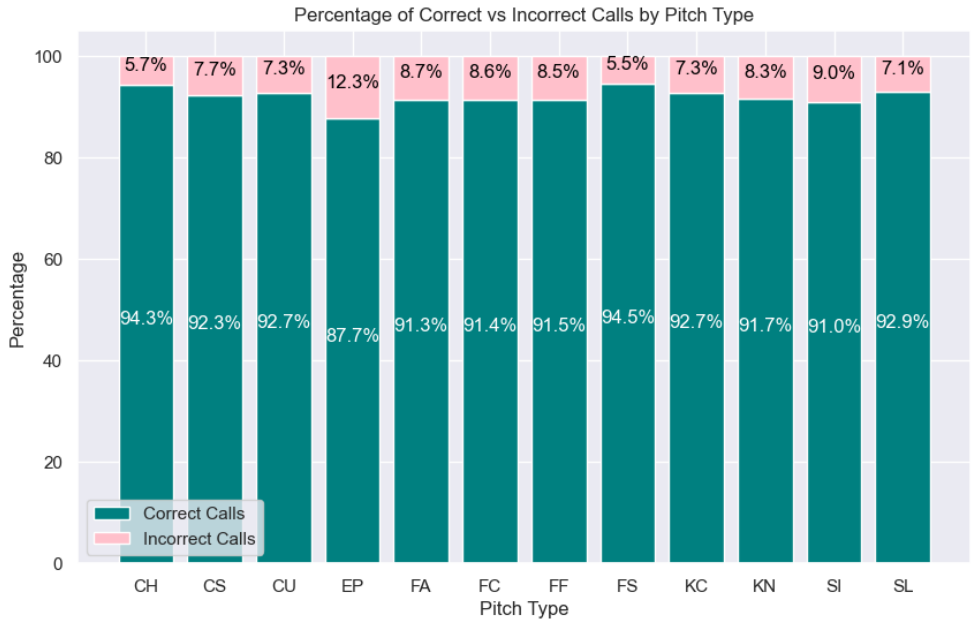
Offspeed



The sample mean deviates from our population mean at our alpha of 0.05. Thus, we reject H0 and accept HA. Therefore, we found an association between correct call percentage and offspeed.

Correct Call Percentage by Pitch Type

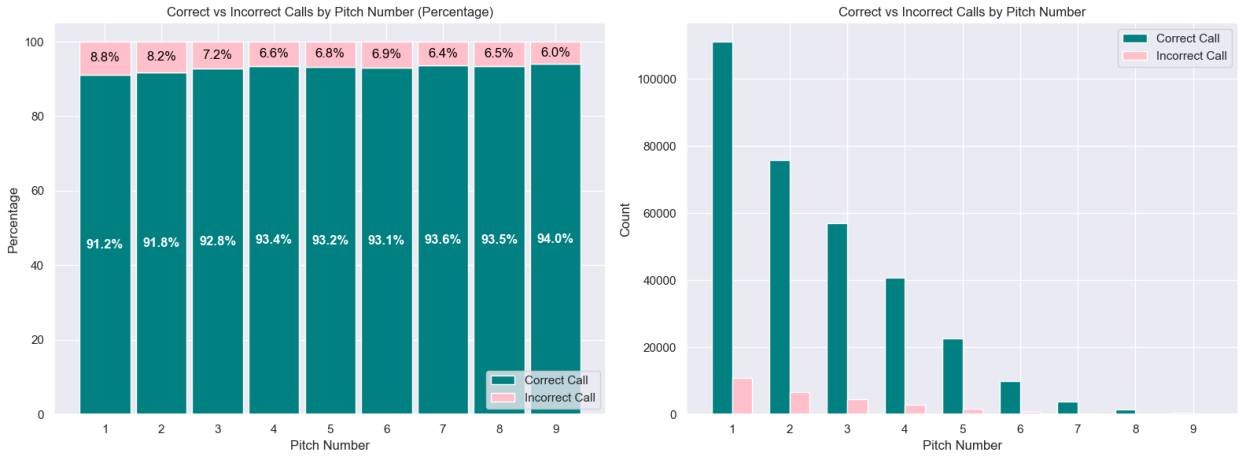
This visualization shows that there is significant variation in correct call percentage depending on the type of pitch that is thrown. This further emphasizes the findings of our t-test.



Correct Call Percentage by Pitch Number

Additionally, we investigated whether pitch count influences the umpire's accuracy in calling strikes and balls. It is possible that umpires become more adept at making correct calls after

observing the same pitcher and batter for several pitches, leading to a better gauging of the strike zone for that specific batter.

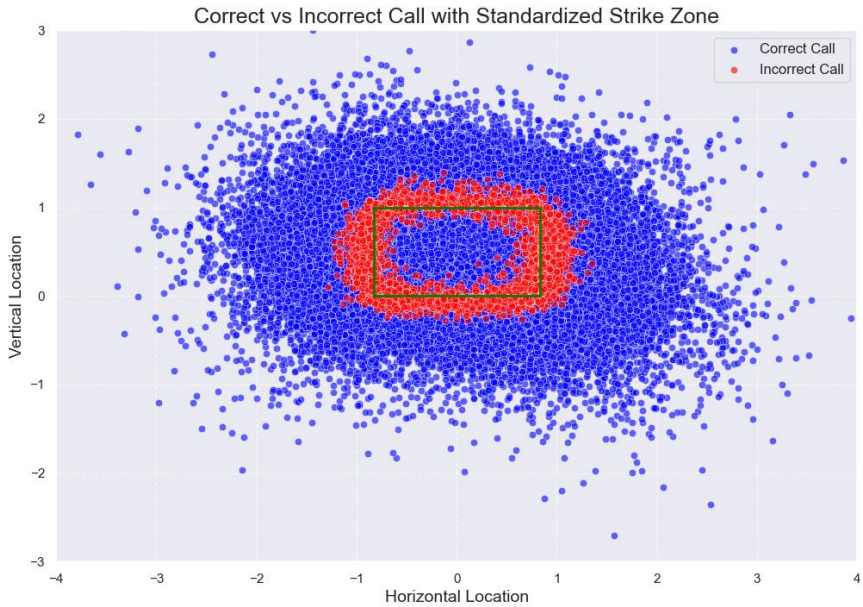


Correct vs. Incorrect Call with Standardized Strike Zone

For each batter, the top and bottom of the strike zone varies due to varying player heights. In order to visualize this, we standardized the strike zone with the formula:

$$Normalized\ Zone = \frac{(Vertical\ Position\ of\ the\ Ball - Bottom\ of\ Strike\ Zone)}{(Top\ of\ Strike\ Zone - Bottom\ of\ Strike\ Zone)}$$

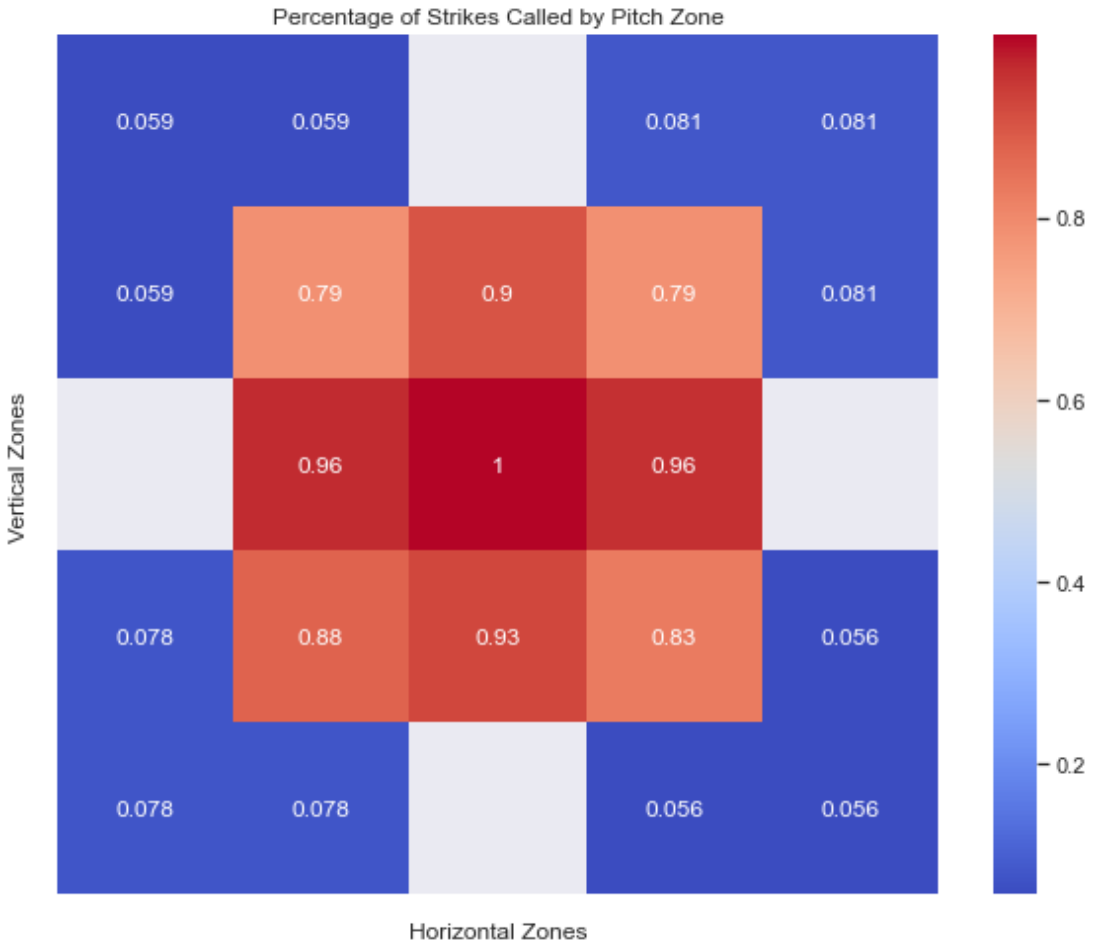
We found that pitches that are close to the edges of the strike zone are frequently called incorrectly, which aligns with what we expected prior to beginning EDA.



Percentage of Strikes Called by Pitch Zone

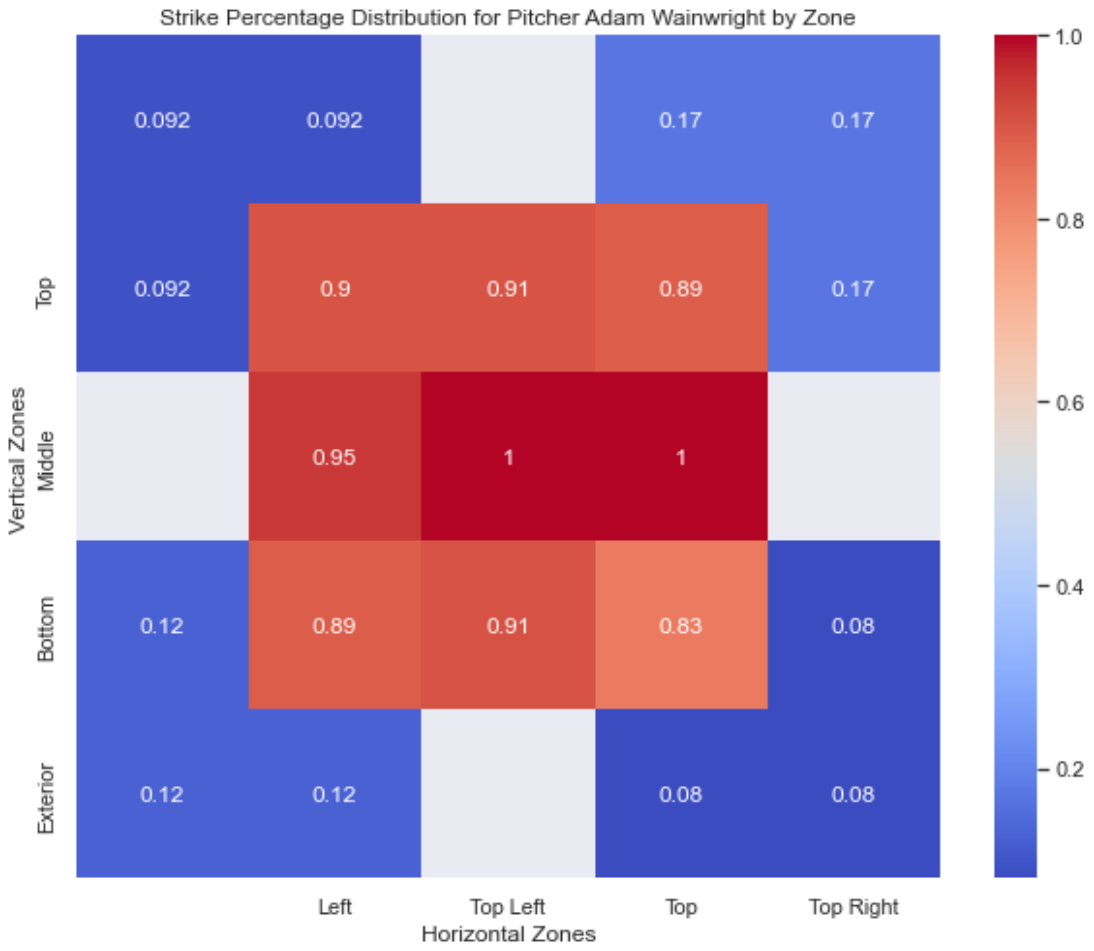
Visualizing strikes called by pitch zone highlights the areas where umpires have the greatest difficulty. With an overall correct call percentage of 92.1%, Umpires are most accurate in calling strikes in the middle zones. However, their accuracy drops by roughly 10% to around 80% in the

corner zones. This suggests that if a batter is uncertain about swinging at a pitch, it might be advantageous not to swing, as there is a higher likelihood of the umpire making an incorrect call in those areas.



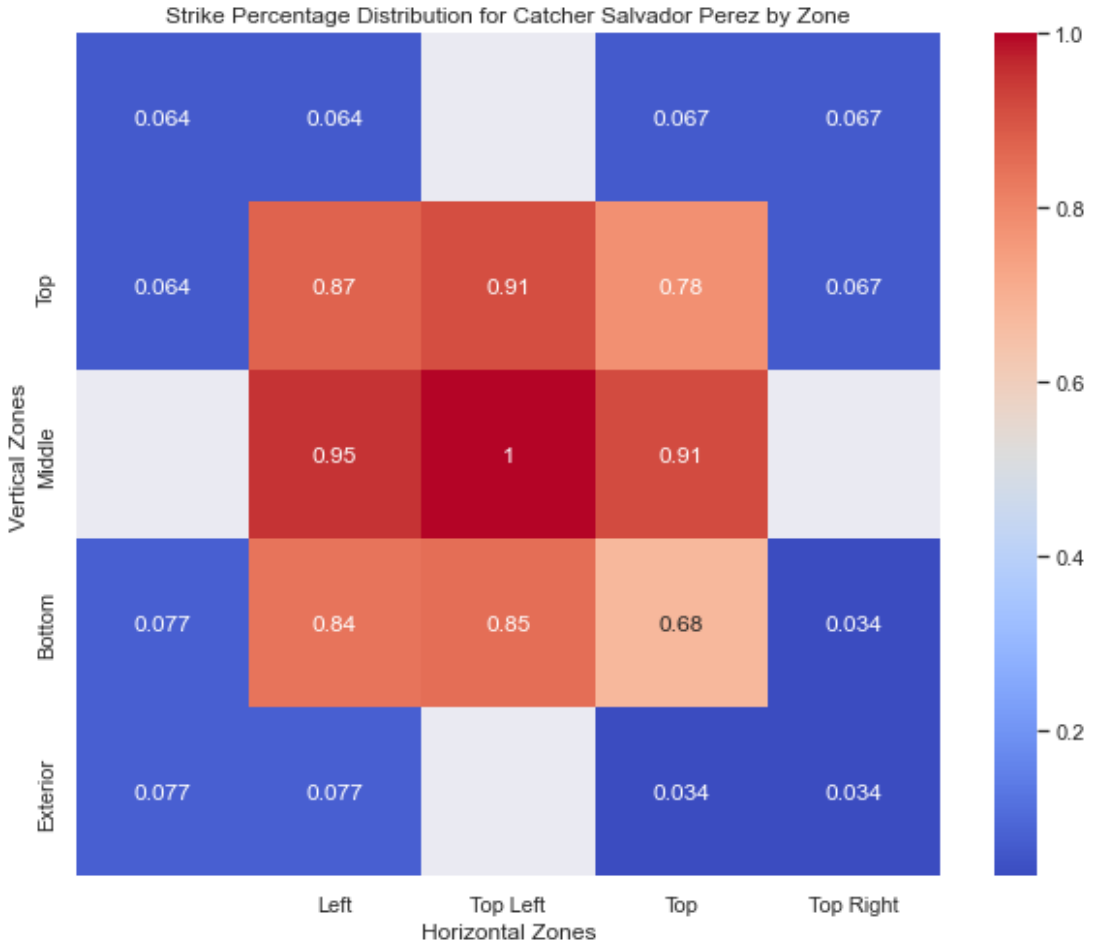
Adam Wainwright

Next, considering the human element in umpire decisions, we hypothesized that umpires might exhibit leniency towards certain pitchers based on their reputation. To investigate this, we analyzed the performance of Adam Wainwright, the pitcher with the most strikes from the St. Louis Cardinals. Interestingly, we found that Wainwright had a significantly higher strike percentage across almost all zones, including in zones where the pitch should have been called a ball. This suggests that his reputation may influence umpire calls, leading to a higher rate of strikes being awarded.



Salvador Perez

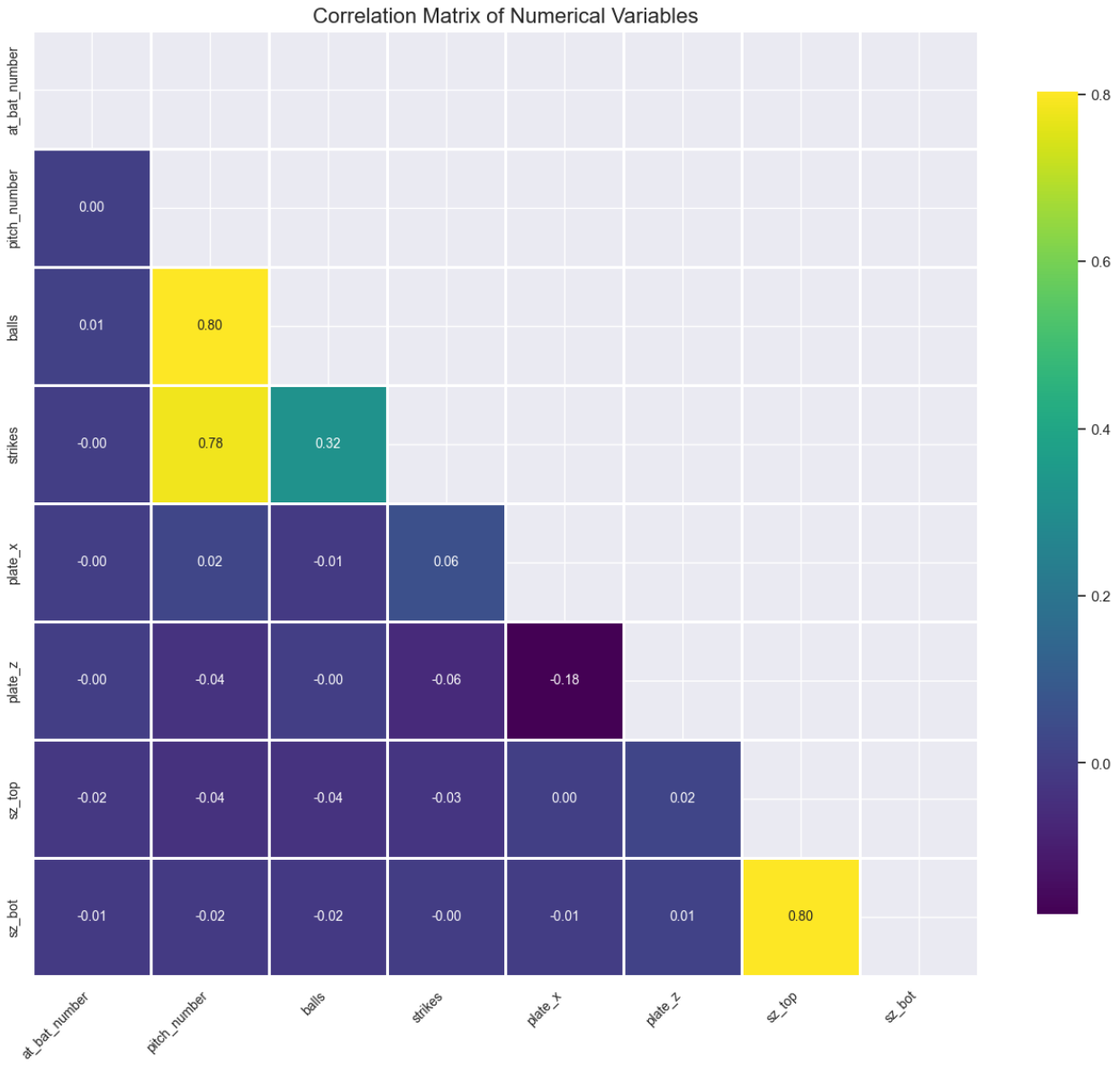
Pitchers, however, only represent half of the equation. The catcher also plays a key role in influencing the umpire's decision by framing the pitch (i.e. subtly moving his glove closer to the strike zone after making a catch). We examined Kansas City Royals All-Star catcher Salvador Perez and found that while his strike percentages in most zones were similar to the league average, he greatly outperformed the league average in zone 1, with an increase of 8%, where most umpires do tend to struggle. This suggests that a combination of pitcher and catcher performance contributes significantly to the umpire's ability to make the correct call.



Correlation Matrix for Numeric Variables

Interestingly, balls and pitch number are very strongly correlated. Strikes and pitch number are as well. This correlation indicates that as more pitches are thrown, there are more opportunities for balls and strikes to be recorded.

Pretty obviously, the bottom and top of the strike zone are highly correlated. This makes sense because typically knee and chest height increase proportionally. However, this does tell us that not everyone's strike zone is the same size.



Modeling:

Selected Variables for Modeling

After using EDA to get a better understanding of our data, we opted to use the following variables to train our models:

- categorical_features: 'zone', 'game_id', 'pitch_type', 'pitcher_id', 'batter_id', 'catcher_id', 'bats_lefty', 'pitches_lefty'
- numerical_features: 'at_bat_number', 'pitch_number', 'balls', 'strikes', 'plate_x', 'plate_z', 'sz_top', 'sz_bot'

Summary of Results

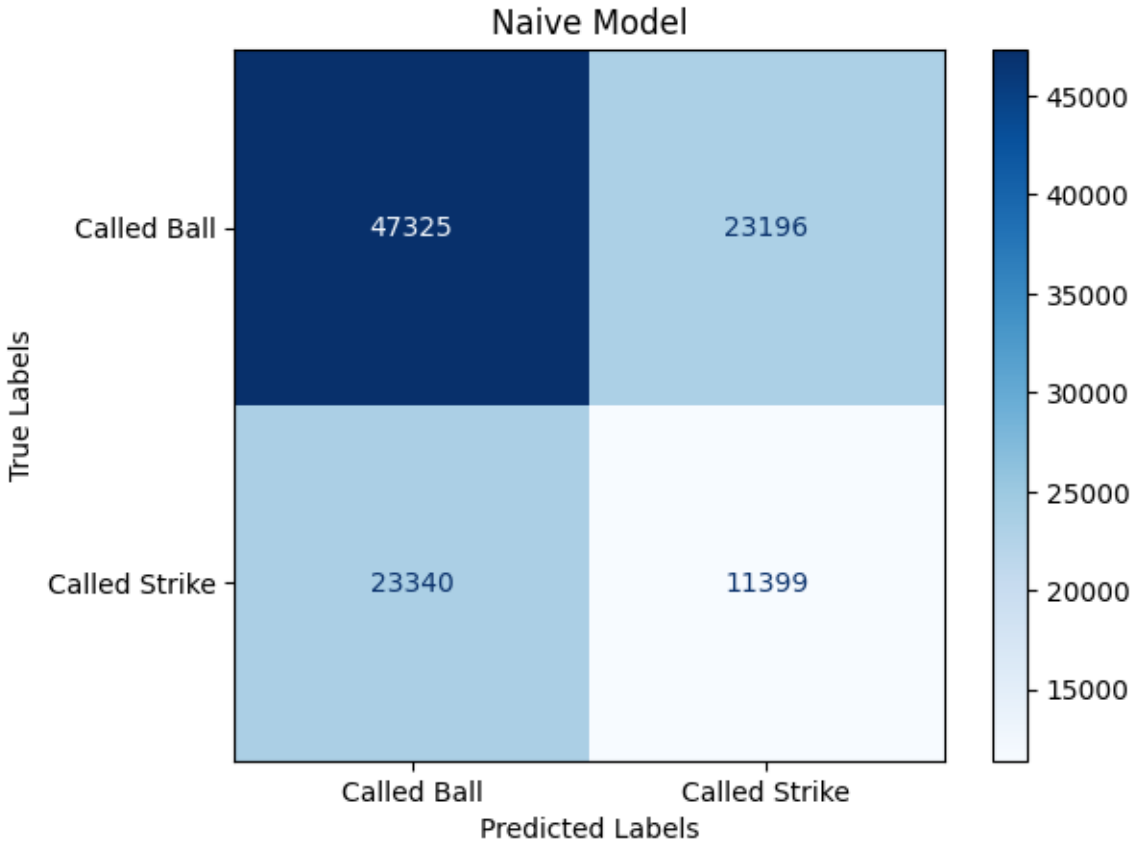
Model	Accuracy	Precision	Recall	F1 Score
Naive	0.56	0.33	0.33	0.33
Logistic Regression (Classifier)	0.92	0.90	0.86	0.88
XGBoost Classifier	0.93	0.88	0.93	0.90
MLP Classifier	0.93	0.90	0.88	0.89

Naïve Model

In order to get a true baseline of our models' performance, we fit a Naive Model running a 'stratified' approach. With this strategy, since 66.99% of the data is a ball, this model will predict ball 66.99% of the time.

Classification Report:

	precision	recall	f1-score	support
Called Ball	0.67	0.67	0.67	70521
Called Strike	0.33	0.33	0.33	34739
accuracy			0.56	105260
macro avg	0.50	0.50	0.50	105260
weighted avg	0.56	0.56	0.56	105260



Logistic Regression

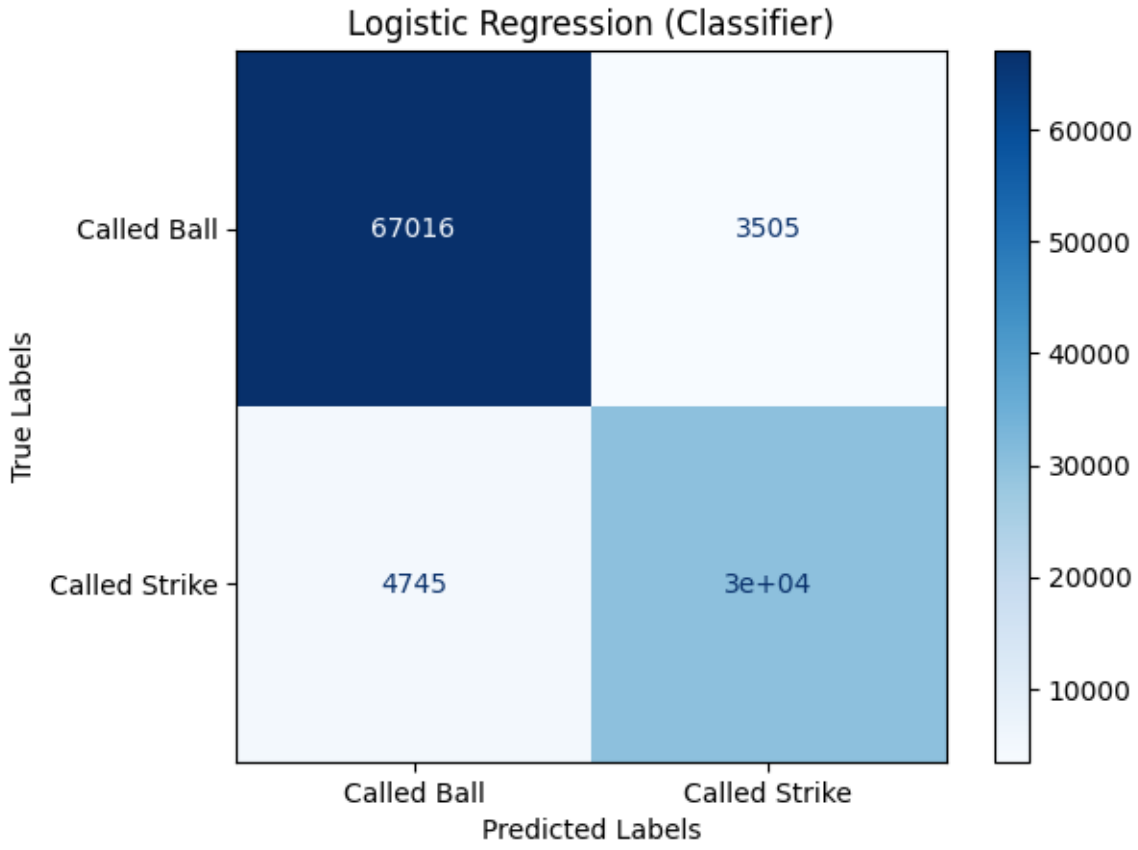
We chose to use this model because it is simple, easy to interpret, and provides a strong baseline for classification.

```
Classification Report:

                precision    recall  f1-score   support

  Called Ball       0.93      0.95      0.94     70521
Called Strike       0.90      0.86      0.88     34739

     accuracy                           0.92    105260
    macro avg       0.91      0.91      0.91    105260
 weighted avg       0.92      0.92      0.92    105260
```

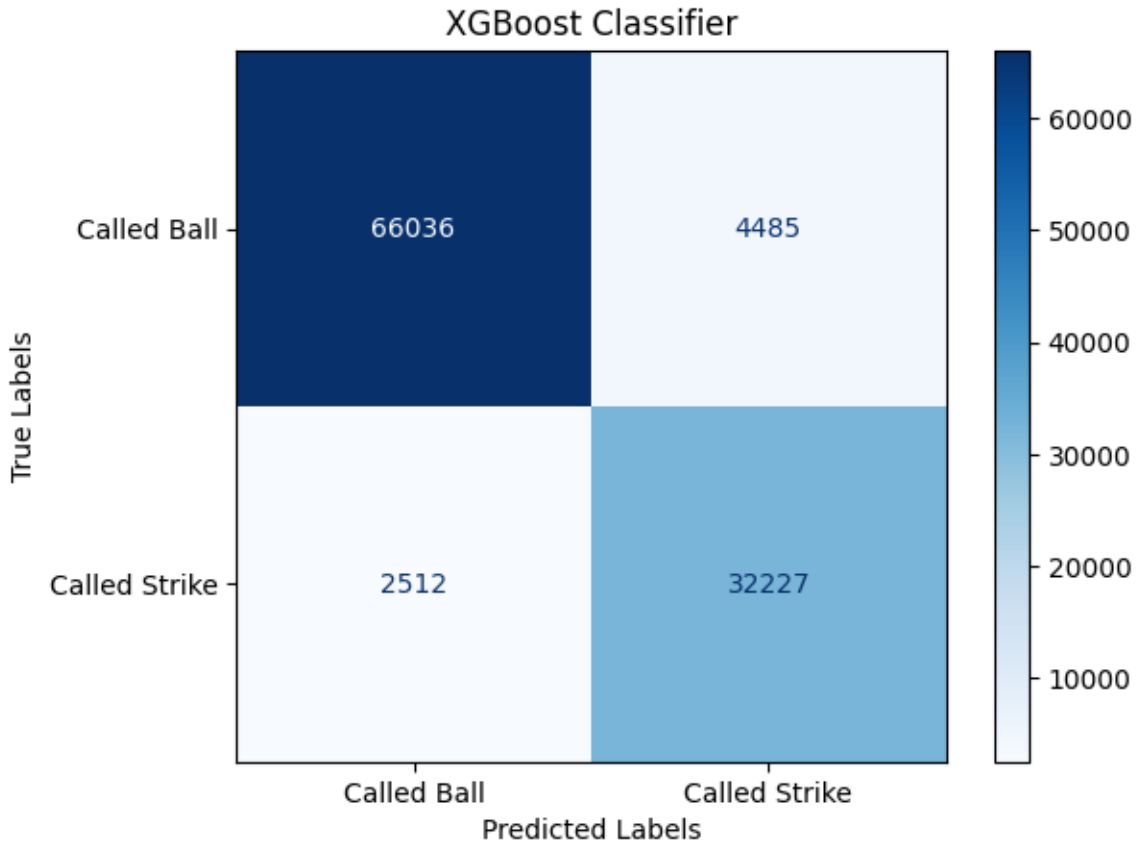


XGBoost Classifier

This model is a good option to use because it is able to handle non-linear data, weigh feature importance, and resist overfitting.

Classification Report:

	precision	recall	f1-score	support
Called Ball	0.96	0.94	0.95	70521
Called Strike	0.88	0.93	0.90	34739
accuracy			0.93	105260
macro avg	0.92	0.93	0.93	105260
weighted avg	0.94	0.93	0.93	105260

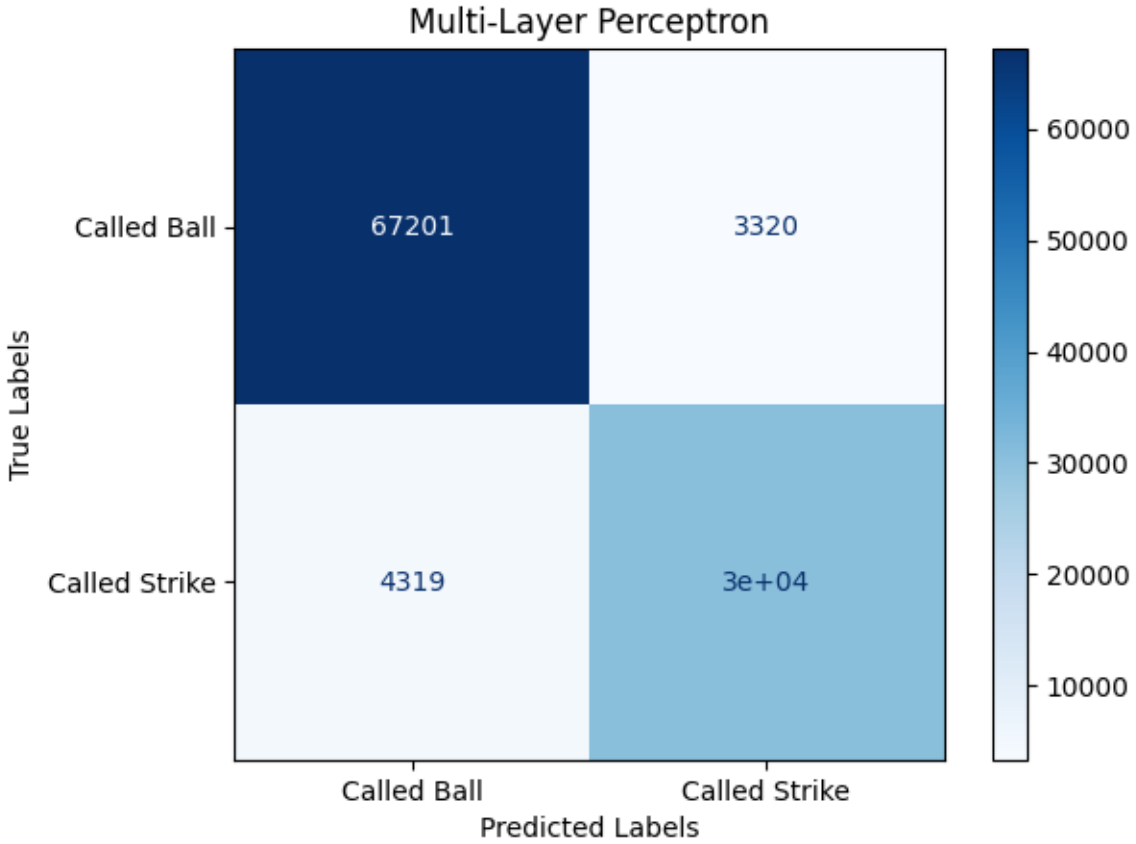


Multi-Layer Perceptron Classifier

Multi-layer Perceptrons are a good option because it can model complex, non-linear relationships. Although it requires more data and computing power, it can provide better results.

Classification Report:

	precision	recall	f1-score	support
Called Ball	0.94	0.95	0.95	70521
Called Strike	0.90	0.88	0.89	34739
accuracy			0.93	105260
macro avg	0.92	0.91	0.92	105260
weighted avg	0.93	0.93	0.93	105260



Best Model

Best Overall Model

The best overall model in terms of a balanced combination of precision, recall, and F1 score is the XGBoost Classifier. It achieves the highest F1 score of 0.90, which indicates a good balance between precision and recall. Although its precision (0.88) is slightly lower than that of the Logistic Regression and MLP Classifiers (both 0.90), its recall (0.93) is the highest among all models, making it the most well-rounded performer.

Best Model for Precision

If the goal is to minimize false positives and prioritize precision, the Logistic Regression Classifier and MLP Classifier are the best choices, both achieving a precision of 0.90. High precision is crucial in scenarios where false positives are costly or undesirable.

Best Model for Recall

If the focus is on identifying as many positive instances as possible, the XGBoost Classifier is the best model with a recall of 0.93. This makes it suitable for applications where missing positive instances (false negatives) is more critical than the occasional false positive.

Best Model for F1 Score

The XGBoost Classifier also has the highest F1 score of 0.90, indicating the best balance between precision and recall. This makes it a strong candidate when both metrics are important, and a trade-off between them is necessary.

Other Considerations

The Naive model performs poorly across all metrics, indicating that it is not a viable option for this classification task. It serves as a baseline for comparison and emphasizes the significant improvements provided by the other models.

Challenges, Limitations, and Recommendations

Challenges

One major challenge in this project is the imbalance in the dataset. 66.99% of the data is labeled as a 'ball'. This imbalance can lead to models being biased towards the majority class and could affect the performance of the model on the minority class.

Another challenge of this project is model interpretability. While advanced models like XGBoost and Multi-Layer Perceptron can offer high accuracy, they are significantly less interpretable compared to simpler models like logistic regression. This can be a challenge when trying to understand the decision-making process of the model or when explaining the model to a non-technical audience.

Limitations

The models are only as good as the data they were trained on. The predictions are based on historical data and may not always be accurate, especially if there are changes in umpiring standards or rules.

These models are also simplified representations of reality. They may not capture all of the complexities of umpire decision making. Factors such as fatigue, weather, and game context may affect decision making.

The models' performance is based on a single dataset used for training and testing. There is no guarantee that the model will perform equally as well on unseen data from different contexts or seasons.

Recommendations

In this project, we used SMOTE to address the imbalanced data problem. Our recommendation is to try oversampling, under sampling, and SMOTE to see what provides the best results.

It may be worthwhile to explore additional features that could influence umpire decisions, such as the score of the game, weather, or number in attendance.

Implement cross validation techniques to all models to ensure that the models' performance is consistent and is not overfitting on the training data. This will generate a more robust model that generalizes well to unseen data.

We recommend to continuously monitor the models' performance and update it with new data to ensure the predictions remain accurate over time. Incorporating recent data can help the model adapt to any changes in umpiring patterns or rules.

Conclusion

After obtaining the data, we completed several rounds of data manipulation to create a new DataFrame encapsulating all the necessary columns for our modeling. After preprocessing, we conducted exploratory data analysis (EDA), which allowed us to get a better understanding of our data. In particular, the EDA enabled us to identify key factors influencing whether a particular pitch would be called a strike, which was our ultimate goal.

Finally, we narrowed down our features to eliminate multicollinearity and tested our data with several machine learning models, including Naive Bayes, Logistic Regression, XGBoost Classifier, and Multi-Layer Perceptron Classifier. Among these, the XGBoost Classifier performed the best due to its high recall and balanced performance across all metrics. The Logistic Regression and MLP Classifiers, however, showed strong results for applications prioritizing precision. Thus, the choice of model should be guided by the specific requirements of the application and the relative importance of precision versus recall. Overall, our MLP and XGBoost models show an improved accuracy in classifying a pitch as a strike or ball in comparison to the umpires.

As mentioned in our recommendations, it would be worthwhile to further investigate our data and incorporate additional variables such as the home plate umpire, game score, weather conditions, and stadium, as these factors could significantly influence the call of a given pitch. While our current model shows promising performance, it could potentially improve with the addition of external data.