Swing Probability Analysis Write-up

Methodology

Data Preparation

I started by loading the data from the three seasons. The data from the first two seasons were combined to create a comprehensive training dataset, while the third season's data was reserved for validation and prediction purposes. Combining the first two seasons helped ensure that the model had enough data to learn the patterns and variations in swing behavior.

Identifying Swing Events

Swing events were identified based on specific pitch descriptions provided in the dataset. I created a binary column `swing` to indicate whether each pitch resulted in a swing. This binary classification was crucial for training a model to predict swing probability.

Feature Engineering

To improve the predictive power of the model, I engineered additional features. This included calculating the height of the strike zone and the distance from the plate center. The strike zone height was determined as the difference between the top and bottom of the strike zone (`sz_top` and `sz_bot`), and the plate distance center was calculated using the Euclidean distance formula on the horizontal (`plate_x`) and vertical (`plate_z`) positions of the pitch. These features provided a more nuanced understanding of the pitch location, which is a significant factor influencing a batter's decision to swing.

Model Training

Categorical variables such as `pitch_type`, `stand`, and `p_throws` were one-hot encoded. This was necessary because machine learning models require numerical input, and the categorical data needed to be converted into a binary matrix as it was relevant to swing probability. The dataset was then split into the training and validation sets. The training set included the standard 80% of data, while the validation set comprised the remaining 20%. This split ensured that the model was trained on a substantial portion of the data while also being validated on an unseen portion to test its performance.

Training the Random Forest Classifier

I chose to use a Random Forest Classifier due to its robustness and ability to handle a large number of features, as were present in this baseball dataset. The model was trained with 500 estimators to ensure a high level of accuracy and reliability. The Random Forest algorithm was selected because it is well-suited for classification tasks and can effectively manage the complexity of baseball swing data.

Model Evaluation

The model was evaluated using accuracy and the ROC-AUC score. The accuracy score measures the proportion of correctly predicted instances, while the ROC-AUC score provides insight into the model's ability to distinguish between classes. The model achieved an accuracy of 87.49% and an ROC-AUC score of 95.08%, indicating strong performance in predicting swing probabilities.

Prediction and Validation

For the third season, I processed the data similarly to the training data. This included calculating the strike zone height and plate distance center, and applying the same one-hot encoding transformation. Ensuring consistency in feature engineering between the training and validation datasets was critical for the model to make accurate predictions. The trained model was used to predict swing probabilities for the third season's pitches. These predictions were appended to the third season dataset as instructed.

Visualizations

The visualizations created for this project offer a detailed look at the swing probabilities for one specific batter (6101) against various pitch types and pitcher handedness combinations. Each scatter plot shows the horizontal (`plate_x`) and vertical (`plate_z`) positions of pitches, with colors representing swing probabilities from low (blue) to high (red). The black box indicates the strike zone, defined by the average top (`sz_top`) and bottom (`sz_bot`) values, with a width of 17 inches (1.417 feet).

Technical Prompts

Prompt 1: Model Evaluation

Evaluation Metrics for Swing Probability Model

To assess the performance of my swing probability model, I utilized two primary evaluation metrics: accuracy and ROC-AUC (Receiver Operating Characteristic - Area Under Curve). These metrics are particularly appropriate for the task of predicting swing probability in baseball for several reasons.

Accuracy: This metric shows the proportion of correct predictions out of all predictions made. In the context of swing probability, accuracy provides a straightforward measure of how often the model correctly predicts whether a batter will swing at a pitch. However, accuracy alone might not be sufficient, especially in scenarios with imbalanced datasets.

ROC-AUC: The ROC-AUC score is a more comprehensive metric that evaluates the model's ability to distinguish between classes (in this case, swing vs. no swing). It measures the area under the ROC curve, which plots the true positive rate against the false positive rate at various

threshold settings. A high ROC-AUC score indicates that the model is good at ranking predictions, making it useful for binary classification tasks.

The accuracy of the model was 87.49%, and the ROC-AUC score was 95.08%. These results suggest that the model performs well in predicting swing probabilities. The high accuracy indicates that the model is correctly predicting swings and non-swings most of the time. Accompanied with the high ROC-AUC score, this demonstrates that the model has a strong ability to differentiate between pitches that will be swung at and those that will not. By analyzing these metrics, I can gain several insights. Both metrics indicate that the model is durable and reliable in its predictions, making it a valuable decision-making tool. Areas for improvement would be highlighted by lower scores, necessitating adjustments.

Communication to Non-Technical Stakeholders: Communicating technical results to non-technical parties is essential. I excel at this due to my background in baseball, first as a player and now as a hybrid front office/player staff member who communicates with coaches, staff, and players daily. The key is to explain the purpose of the analysis, highlight the relevant information, and clarify what it means specifically to them. Visual aids can be particularly effective in these situations. Building rapport within the organization also facilitates productive communication. Ultimately, the research is pointless if the insights are not properly communicated. An Analyst's role is about serving other departments and implementing this mindset to open the door to productive communication.

Prompt 3: Model Selection and Tuning

Model Selection: Random Forest Classifier

For predicting swing probability, I chose the Random Forest Classifier because of its ability to handle a large number of features and complex interactions. The Random Forest algorithm is well-suited for binary classification tasks like this because it builds multiple decision trees and combines their outputs to improve predictive performance and reduce overfitting.

Strengths: The strengths of using the Random Forest Classifier include its robustness against overfitting compared to individual decision trees. It can effectively manage both numerical and categorical features, making it ideal for the diverse data in this project. The algorithm also provides insights into feature importance, helping identify which features are most influential in predicting swing probabilities.

Limitations: Using a Random Forest Classifier can be computationally intensive. Training a Random Forest model with a large number of estimators can be time-consuming. To mitigate this, I saved the model for reproducibility, so it could be reloaded without needing to rerun every time the kernel was restarted. Understanding the combined decisions of numerous trees can also be challenging.

Hyperparameter Tuning: To optimize model performance, I tuned the hyperparameters using grid search cross-validation. Key hyperparameters included the number of estimators (`n_estimators`), the maximum depth of the trees (`max_depth`), and the minimum samples

required to split a node (`min_samples_split`). I selected a Random Forest with 500 estimators to balance performance and computational efficiency. My rationale for the parameters is as follows:

- `n_estimators=500`: Increasing the number of trees typically improves performance but also increases computation time. 500 trees provided a good balance between accuracy and training time.
- `max_depth=None`: Allowing the trees to grow to their full depth enabled the model to capture complex patterns in the data.
- `min_samples_split=2`: Setting this parameter to its default value ensured that the model could capture fine-grained patterns in the data.

Tuning these hyperparameters helped improve the model's generalization ability and predictive performance. The selected parameters ensured that the model was both accurate and efficient, making it highly useful in a baseball analytics context.

Technical Applications

Explaining Important Variables to a Coach

We define middle-middle pitches as those that are within 6 inches of the center of the strike zone in every direction. These pitches are prime targets for swings, contact, and damage. The key variables in predicting swing probability for these pitches are `plate_x`, `plate_z`, and `release_speed`. The `plate_x` and `plate_z` variables represent the horizontal and vertical positions of the pitch as it crosses the plate. For middle-middle pitches, both values are close to zero, indicating the pitch is centered in the strike zone. The more middle a pitch is, the more likely a batter is to swing at it. Additionally, pitch velocity plays a crucial role. Higher velocity pitches tend to have higher swing probabilities because batters have less time to react. In contrast, slower pitches might be taken more often, especially if they have significant movement or speed differential from the fastball. The central location and speed of the pitch are the most important factors influencing swing probability for middle-middle pitches.

Metric: Swing Probability Above Average (SPAA)

To compare individual players to the league average, I created the Swing Probability Above Average (SPAA) metric. This metric measures the difference between a player's swing probability and the league average swing probability, aggregated over all pitches in Season 2. The methodology involves three main steps. First, I calculated the league average swing probability for all pitches in Season 2, which serves as the baseline for comparison. Next, for each player, I calculated their swing probability for all pitches they faced in Season 2. Finally, for each player, I subtracted the league average swing probability from their individual swing probability to compute the SPAA. This gives us a measure of how much more or less likely a player is to swing compared to the league average.

For example, after calculating the SPAA for all players in Season 2, I identified the top 10 and bottom 10 players based on this metric. The top 10 players by SPAA are those who swing more

frequently than the league average, while the bottom 10 players swing less frequently. This metric provides a clear and straightforward way to evaluate each player's swing tendencies compared to the league average. Coaches and analysts can use this metric to identify players who are more aggressive or more selective at the plate, informing decisions on batting order, training focus, and game strategy.

League Average Swing Probability: 0.477
SPAA value range: -0.47728286335281 to 0.3004949144249678

Top 10 Players by SPAA:

1. 5240: 0.300
2. 6429: 0.215
3. 6561: 0.189
4. 5325: 0.176
5. 6210: 0.173
6. 6854: 0.168
7. 6831: 0.159
8. 7122: 0.156
9. 6039: 0.155
10. 6847: 0.151

Bottom 10 Players by SPAA:

1. 5803: -0.477
2. 5890: -0.477
3. 6783: -0.227
4. 7057: -0.201
5. 6497: -0.183
6. 6274: -0.170
7. 7042: -0.161
8. 5696: -0.160
9. 5569: -0.156
10. 5376: -0.144