

Transformer-Based Baseball Modeling for Pitch Outcome Prediction and Strategy Optimization

Declan Kneita

1. Introduction

The intricate duel between pitcher and batter lies at the heart of baseball, where each pitch can significantly alter the trajectory of a game. Predicting the outcome of individual pitches is a fundamental challenge with profound implications for optimizing defensive alignments, pitch sequencing, and overall game strategy. Traditional analytical methods have predominantly relied on aggregate statistics or heuristic strategies, such as evaluating player-specific batting averages by pitch type or adhering to conventional pitching philosophies like "hard in, soft away" [10]. While these approaches offer some insights, they often lack the granularity and adaptability needed to capture the rich context and temporal dependencies inherent in sequences of pitches.

Advancements in machine learning have opened new avenues for modeling baseball at a finer scale. By leveraging extensive play-by-play data, recent research has employed sophisticated models to predict outcomes and inform strategic decisions [2, 3, 9]. However, many of these approaches focus on modeling the game at the level of at-bats or use static player embeddings, which may not generalize well to unseen players or adapt quickly to changing game contexts. Models operating at the at-bat level are limited in providing actionable insights for individual pitches, thereby restricting their utility in real-time decision-making where pitch-by-pitch analysis is crucial.

To address these limitations, we propose a novel transformer-based neural network model designed specifically for predicting the outcomes of individual pitches [4]. Our model captures intricate relationships within sequences of pitches by learning dependencies such as how previous pitches influence current outcomes and how batters' recent tendencies inform their likely responses. Unlike methods relying on static player embeddings or aggregate statistics, our approach is dynamic and adaptable, capable of learning general trends in pitch outcomes while applying to any player with sufficient pitch context. This adaptability enables real-time deployment and application to players beyond those included in the training data.

Furthermore, our model not only predicts the result of a pitch but also forecasts the likely hit location should contact occur. This dual predictive capability provides comprehensive insights that can inform both pitching strategies and defensive alignments.

2. Related Work

Baseball analytics has increasingly embraced advanced machine learning techniques to deepen the understanding and prediction of player performance and game outcomes. While traditional sabermetrics offer foundational metrics, recent research endeavors have aimed to capture the nuanced and contextual aspects of player actions through sophisticated modeling approaches.

Heaton and Mitra have significantly contributed to this field through their application of transformer-based models to player performance prediction. In their 2021 study, "Using Machine Learning to Describe How Players Impact the Game in the MLB," they explored the use of machine learning for player description and game outcome forecasting [1]. Employing a transformer-based architecture inspired by Natural Language Processing and Computer Vision, they created embeddings that reflect a player's influence over sequences of plate appearances. These embeddings provided a nuanced understanding of player performance, differentiating between various impact factors such as pitch sequencing and situational performance, thereby enhancing predictive tasks like game outcome prediction.

Building upon this foundation, their 2023 paper, "Learning Contextual Event Embeddings to Predict Player Performance in the MLB," introduced a transformer-based model that generates 64-dimensional embeddings to capture a player's short-term and long-term impact on the game [2]. By leveraging detailed play-by-play data and integrating contextual information such as pitch type, location, and game state, their model moved beyond traditional counting statistics. This approach not only provided a more dynamic representation of player performance but also achieved competitive prediction accuracy with major sportsbooks, all while making a larger number of predictions.

In a different vein, Melville et al. (2023) presented a distinct approach in their paper, "A Game Theoretical Approach to Optimal Pitch Sequencing" [3]. They modeled the pitcher-batter interaction as a zero-sum game, seeking equilibrium strategies that optimize pitch sequencing to minimize the expected run value. By exploring three different game models with varying levels of information available to the batter, they incorporated strategic decision-making into pitch selection. Their introduction of OptimusPitch, a recurrent neural network designed to predict pitch outcomes based on sequential pitch data, integrated game theory with machine learning to provide actionable insights for optimal pitch sequencing.

3. Methods

Building upon these foundations, our research introduces a transformer-based neural network specifically tailored for single-pitch outcome prediction. Unlike Heaton and Mitra's broader player embedding models, our approach focuses on granular, real-time predictions of pitch outcomes by modeling sequences of 400 consecutive pitches. This fine-grained focus allows us to capture intricate temporal dependencies and contextual factors that influence each pitch's outcome, while

providing actionable insights for immediate strategic optimizations such as pitch sequencing and defensive alignments.

Moreover, while Melville et al. integrate game theory with recurrent neural networks to derive optimal pitching strategies, our model operates independently of a strategic game-theoretic framework. Instead, we prioritize enhancing predictive accuracy at the single-pitch level, enabling real-time decision-making without the need for equilibrium computations. This distinction allows our model to be more flexible and applicable in dynamic in-game scenarios where immediate pitch outcome predictions are crucial.

3.1. Data

The dataset for this study was sourced from the Statcast database and collected using the pybaseball Python package [5, 6]. It comprises pitch-by-pitch data from every Major League Baseball (MLB) game between 2015 and 2024. Data from 2015 to 2022 was utilized for model training and validation, while data from 2023 to 2024 was reserved for testing purposes. Each record in the dataset corresponds to a single pitch and includes detailed information about the pitch itself, its outcome, the players involved, and the game context.

For the single-pitch prediction task, sequences of pitches were constructed as input to the model, where each pitch is represented by a combination of continuous and categorical features that describe various aspects of the pitch, its result, and the game state. The continuous features include measurements such as pitch velocity, spin rate, exit velocity, launch angle, hit coordinates, and release point coordinates. The categorical features encompass the pitch result, pitch type, hit location, game state (including ball count, strike count, out count, base occupancy, and inning number), batter and pitcher handedness, and zone location. Meta-information such as player names and game identifiers were used solely for constructing sequences and were not included in the model's input features.

In preparing the data for modeling, continuous features were standardized to have zero mean and unit variance to ensure uniform scaling and facilitate model training. Categorical features were transformed using one-hot encoding, converting them into binary vector representations suitable for input into the neural network. To reduce sparsity and improve computational efficiency, rare pitch types (e.g., intentional walks) and uncommon events (e.g., pickoffs) were excluded from the dataset. Additionally, semantically similar outcomes, such as "Grounded into Double Play" and "Fielder's Choice," were consolidated under the "Field Out" category to streamline the classification task.

Each pitch is thus represented as an 87-dimensional feature vector, encompassing all continuous features and the expanded categorical features after one-hot encoding (Figure 1). In addition, mask dimensions are added for features related to the outcome of a pitch, these dimensions will be detailed further in 3.4.2. Input to our single-pitch prediction transformer model is structured as a sequence of these pitch feature vectors.

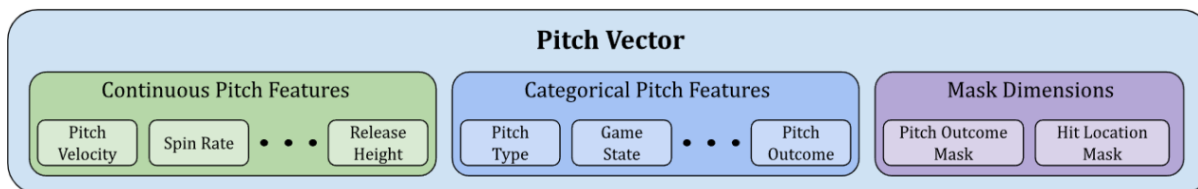


Figure 1: Pitch Feature Vector

3.2. Sequence Modeling

To create pitch sequences, all pitches are grouped by batter and ordered chronologically by game date, at-bat number, and pitch number—reconstructing the temporal sequence of events. Sequences of 400 consecutive pitches are then extracted for each player using a sliding window with a one-pitch overlap. Window sizes of 200, 400, and 600 pitches were experimented with, with 400 pitches performing the best based on validation accuracy and computational efficiency. Each sequence represents a single training example, with subsequent sequences shifting the window forward by one pitch. For instance, a batter with 401 pitches in their history yields two sequences: the first spanning pitches 1-400 and the second spanning pitches 2-401.

3.3 Model Architecture

The architecture of the model used in this research leverages transformer-based networks, a class of models widely used for processing sequential data due to their capacity to capture intricate dependencies between elements within a sequence. In this section, we provide a detailed overview of the transformer foundation and how it is adapted in our model to predict pitch outcomes with high precision.

3.3.1 Transformer Fundamentals

Transformers process sequences by employing a self-attention mechanism, which dynamically assigns weights to each element based on its importance relative to others in the sequence. This capability enables the model to focus on critical interactions across pitches, regardless of their position within the sequence. To encode the sequential order of the data, transformers utilize positional encodings, which provide information about the temporal structure of the input. This architecture has proven highly efficient, especially when handling long sequences, due to its ability to process data in parallel [4].

3.3.2 Single-Pitch Prediction Transformer

Our model builds on this foundation, specifically tailoring the transformer architecture to perform single-pitch result prediction. Each input sequence consists of 400 consecutive pitches, with each pitch represented as an 87-dimensional vector. The initial step in the model pipeline involves projecting the input vectors into a higher-dimensional space using a linear embedding layer. This

transformation produces rich feature representations, enhancing the model's ability to learn complex relationships.

To incorporate positional information into the sequence, sinusoidal positional encodings are added to the embedded features [4]. This step ensures that the model recognizes the order of pitches within the sequence, which is critical for capturing temporal dependencies.

The core of the architecture consists of a 12-layer transformer encoder (Figure 2). Each layer integrates multi-head self-attention mechanisms and feedforward neural networks. The self-attention mechanism allows the model to weigh the relevance of different pitches in the sequence, identifying patterns and dependencies that span multiple games or player interactions. Residual connections and layer normalization are incorporated into each layer to stabilize training and facilitate gradient flow [4].

A distinctive feature of the model is its handling of the final pitch in the sequence. While the transformer encoder processes the entire sequence, the original features of the last pitch are retained and passed through a linear transformation. This transformed representation is then concatenated with the encoder's output for the final pitch. This residual connection ensures that critical information about the last pitch is preserved, complementing the contextual embeddings generated by the encoder.

The concatenated features are passed through two fully connected layers with ReLU activations, refining the combined representation and producing the final predictions. In total, our model uses roughly 800,000 parameters. The model outputs a 24-dimensional vector, representing predictions for multiple tasks. Among these, two primary tasks are the prediction of pitch results and hit locations, both modeled as probability distributions. The pitch result is captured by a 10-class distribution, encompassing outcomes such as strikes, balls, singles, and home runs. Similarly, the hit location prediction includes nine classes, corresponding to fielder positions such as left field, shortstop, and right field.

The remaining dimensions of the output vector correspond to predictions for several continuous features, such as launch angle, exit velocity, and hit coordinates. These continuous predictions were included as part of the model's multi-task learning framework, designed to improve the overall performance by encouraging the model to learn a broader set of interdependent tasks. However, upon evaluation, the predictions for these continuous features were found to be less accurate and ultimately not useful for practical applications. As a result, they are not utilized in the final model outputs or downstream analyses, though their inclusion during training may have contributed to the model's generalization capabilities.

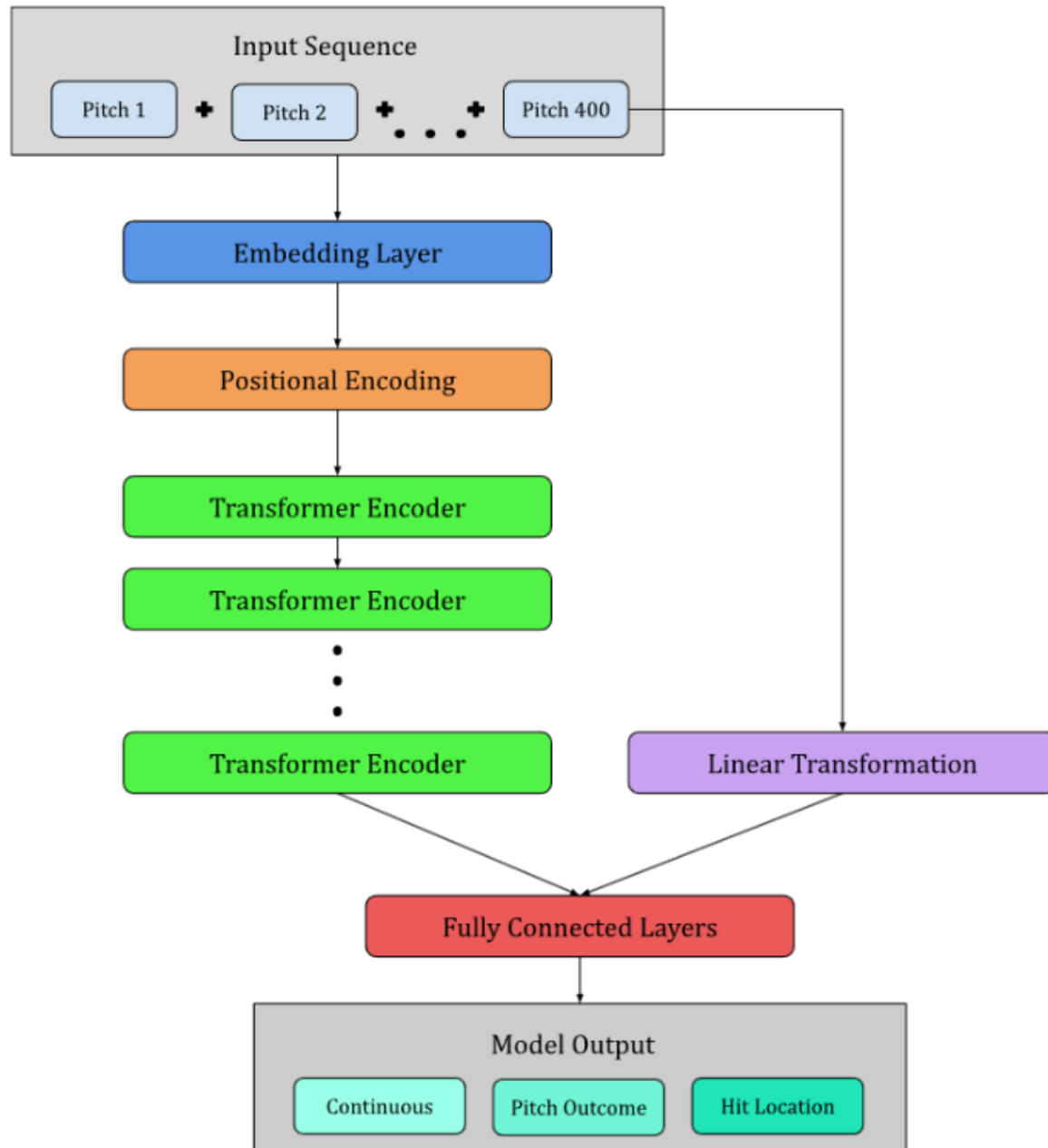


Figure 2: Single-Pitch Prediction Transformer Architecture

3.4 Model Training

The model training process leverages a multi-task learning framework to predict both categorical and continuous features of pitch outcomes [11]. This framework enables the model to learn representations for multiple related tasks simultaneously, enhancing its understanding of baseball dynamics and supporting accurate single-pitch prediction.

3.4.1 Single Pitch Prediction

Single pitch prediction refers to the model's ability to infer the outcome of a single pitch within a sequence, specifically the final pitch of a 400-pitch sequence. The task involves predicting the true values of continuous result features like launch angle and exit velocity and generating probability distributions for two key outcome dimensions: pitch result (e.g., strike, ball, single) and hit location (e.g., shortstop, center field). These predictions rely on the observed characteristics of the pitch itself, and the context provided by preceding pitches in the sequence.

3.4.2 Sub-Token Masking

To allow for single pitch prediction during training and inference, the model applies sub-token masking to the final pitch in each sequence. This process selectively obscures outcome-related features while preserving observable characteristics of the pitch, ensuring the model cannot directly access the true result.

- **Masked Features:** Outcome dimensions such as pitch result, hit location, launch speed, and hit coordinates are masked. For continuous features, the true values are replaced with their global mean values. For categorical features, the true value in the one-hot encoding is set to zero, and a "masked" dimension is activated to signal that the feature must be inferred.
- **Unmasked Features:** Pitch-specific characteristics, including velocity, spin rate, and release point, remain unmasked. These features reflect real-world information available during a game and provide the foundation for outcome predictions.

By leveraging sub-token masking, the model learns to infer unknown outcomes based on both the observable features of the final pitch and its surrounding context within the sequence.

3.4.3 Loss Function

The multi-task loss function combines objectives for categorical and continuous predictions, reflecting the dual nature of the model's learning goals. The total loss is defined as:

$$L_{Total} = 0.7(L_{PitchOutcome} + L_{HitLocation}) + 0.3L_{Continuous}$$

where:

- $L_{PitchOutcome}$ and $L_{HitLocation}$: Cross-entropy losses for categorical predictions [12]. These dominate the total loss, emphasizing the importance of accurate pitch result and hit location predictions.
- $L_{Continuous}$: Mean squared error (MSE) loss for continuous predictions, such as launch speed and hit coordinates [13]. These auxiliary dimensions enrich the model's understanding without detracting from the primary objectives.

The loss function prioritizes categorical predictions, ensuring the model excels at the core single-pitch prediction tasks. The auxiliary continuous losses act as supplemental learning signals, improving the model's representation of pitch dynamics.

3.4.4 Training Procedure

The model is trained on sequences of 400 pitches constructed from MLB games spanning 2015 to 2022. For each sequence, the final pitch serves as the target for single-pitch prediction. Outcome features for the final pitch are masked, and the model learns to predict these masked values using the observed features of the pitch and the broader sequence context. By minimizing the multi-task loss, the model refines its ability to make accurate single-pitch predictions while generalizing effectively to new scenarios.

4. Results

This section presents a comprehensive evaluation of our transformer-based model's performance in predicting pitch outcomes and optimizing pitch sequencing strategies. We compare our model against two baselines: a historical average baseline and an XGBoost model trained on the same pre-2023 data. The evaluations focus on the models' abilities to rank probable outcomes effectively despite inherent class imbalances, adapt to different player types with distinct hitting tendencies, and identify optimal pitches in specific game scenarios.

We begin by describing the models used in the evaluation:

Historical Average Baseline: This baseline computes the mean distribution of pitch outcomes using data from the 2022 season and predicts this same distribution for every pitch in the 2023 season. It serves as a naive benchmark, lacking adaptability to specific game contexts or player tendencies.

XGBoost Model: An XGBoost classifier is trained on the same feature set used for the transformer model but operates on individual pitches without sequential context. XGBoost is known for its effectiveness in handling structured data and capturing nonlinear relationships, providing a robust baseline for comparison [14].

Transformer-Based Model: Our proposed model utilizes sequences of 400 pitches to predict single-pitch outcomes, leveraging its ability to model sequential dependencies and contextual nuances within the game. By incorporating the sequential nature of baseball events, the transformer model aims to provide more accurate and context-sensitive predictions.

The following subsections detail the evaluation of these models using top-k precision metrics, demonstrate the transformer model's adaptability through player-specific analyses, and illustrate its practical utility with an optimal pitch selection framework applied in a case study.

4.1 Top-K Precision Metrics

To evaluate the models' abilities to prioritize the most probable and strategically significant outcomes, we employed top-k precision metrics, focusing specifically on top-4 precision. This metric assesses the proportion of times the true outcome is among the top-k predicted probabilities for each pitch, providing insights into the models' effectiveness in ranking likely outcomes.

4.1.1 Rationale for Choosing Top-4 Precision

The choice of $k = 4$ is motivated by the class imbalance inherent in baseball pitch outcomes. Since events like 'Ball' and 'Strike' dominate, traditional accuracy metrics do not adequately capture the models' performance on less frequent but crucial events. By considering the top four predicted outcomes, we ensure that both common and significant rare events are included, providing a balanced evaluation of the models' predictive capabilities.

4.1.2 Results

For each pitch in the 2023 season test set, each model generated a probability distribution over possible pitch outcomes and hit locations. The top four outcomes with the highest predicted probabilities were selected as the model's predictions for that pitch. The top-4 precision for each outcome class was then calculated as the proportion of times the true outcome appeared within the model's top four predictions. The historical average baseline in this case makes predictions by randomly sampling 4 events using the historical distribution from the 2022 season. The top-4 precision metrics for each model can be seen in Tables 1 and 2.

Outcome	Historical Average	XGBoost Baseline	Transformer Model
Ball	0.96	0.99	0.99
Strike	0.96	0.99	0.99
Double	0.13	0.3	0.43
Field Out	0.76	0.99	0.99
Hit by Pitch	0.02	0.95	0.96
Home Run	0.09	0.23	0.34
Single	0.35	0.86	0.97
Strikeout	0.48	0.99	0.99
Triple	0	0	0
Walk	0.2	0.99	0.99

Table 1: Top-4 Precisions for Pitch Outcomes

For common outcomes such as 'Ball' and 'Strike', both the transformer and XGBoost models achieved exceptionally high top-4 precision scores, exceeding 99%. The randomized baseline performed significantly worse, highlighting the importance of model-driven predictions over naive approaches.

For less frequent but strategically important outcomes, the transformer model demonstrated a notable advantage. For 'Single', the transformer achieved a top-4 precision of 97%, outperforming the XGBoost model's 86% and the randomized baseline's 35%. Similarly, for 'Double' and 'Home Run', the transformer model achieved precisions of 43% and 34%, respectively, compared to 30% and 23% for XGBoost, and substantially lower scores for the randomized baseline.

Hit Location	Historical Average	XGBoost Baseline	Transformer Model
Pitcher	0.13	0.05	0.14
Catcher	0.04	0.06	0.12
First Base	0.24	0.31	0.32
Second Base	0.38	0.45	0.53
Third Base	0.32	0.51	0.52
Shortstop	0.41	0.56	0.6
Left Field	0.47	0.77	0.78
Center Field	0.51	0.83	0.83
Right Field	0.48	0.76	0.73

Table 2: Top-4 Precisions for Hit Locations

For positions like 'Shortstop', 'Second Base', and 'Third Base', the transformer model achieves top-4 precisions of 60%, 53%, and 52%, respectively. These are higher than the XGBoost model's scores of 56%, 45%, and 51%, and significantly exceed the historical baseline's scores. Accurate predictions in these positions are vital for infield defensive alignments and can influence decisions on player positioning and shifts.

The transformer model also performs well in predicting hits to outfield positions. It achieves top-4 precisions of 78% for 'Left Field', 83% for 'Center Field', and 73% for 'Right Field'. These scores are comparable to or slightly lower than the XGBoost model's scores for 'Center Field' and 'Right Field' but still represent strong predictive performance. The historical baseline lags behind, highlighting the advantage of the transformer model in leveraging contextual information.

For less common hit locations like 'Pitcher' and 'Catcher', the transformer model shows a significant improvement over the XGBoost model and the historical baseline. Although the absolute precision

percentages are lower due to the rarity of these events, the transformer model's ability to better predict these outcomes demonstrates its nuanced understanding of player tendencies and pitch contexts

The transformer-based model's superior performance in top-4 precision metrics, particularly for less frequent but impactful events like 'Double' and 'Home Run', underscores its ability to adapt predictions based on the specific context of each pitch. While both the transformer and XGBoost models performed similarly on common outcomes ('Strikeout' and 'Field Out'), the transformer model showed a significant advantage in predicting less frequent events, which are crucial for strategic decision-making.

4.2 Model Adaptability to Player Tendencies

To evaluate the models' abilities to adapt predictions based on specific player tendencies, we conducted a comparative analysis using two players with distinct hitting profiles: Aaron Judge, a well known power hitter, and Luis Arraez, a premier contact hitter. By comparing the aggregated stat predictions from both the transformer-based model and the XGBoost baseline against the actual statistics from the 2023 season, (Both models have only been trained on data prior to the 2023 season), we assessed how well each model captures the unique characteristics of these players.

For each player, we used both models to predict the outcome probabilities for every pitch they faced during the 2023 season. Summing these probabilities across all pitches provided the expected totals for each event category. This approach allowed us to evaluate how effectively each model adapts to individual player tendencies without explicit player identifiers in the input data.

Aaron Judge faced a total of 1,612 pitches in the 2023 season. Table 3 presents the predicted and actual event counts.

Outcome	XGBoost Predicted	Transformer Predicted	Actual Count
Ball	639.81	630.09	619
Strike	634.66	586.32	603
Double	11.04	15.66	14
Field Out	101.44	126.09	121
Hit by Pitch	0.51	0.26	0
Home Run	13.94	25.6	32
Single	37.97	42.85	36

Strikeout	107.5	105.76	107
Triple	0.62	0.91	0
Walk	64.5	67.46	69

Table 3: Predicted vs. Actual Event Counts for Aaron Judge

Luis Arraez faced a total of 1,848 pitches in the 2023 season. Table 4 shows the predicted and actual event counts.

Outcome	XGBoost Predicted	Transformer Predicted	Actual Count
Ball	614.31	478.08	484
Strike	812.71	677.65	663
Double	19.27	24.09	21
Field Out	205.44	256.61	263
Hit by Pitch	2.19	2.07	3
Home Run	12.59	6.27	9
Single	64.54	100.42	116
Strikeout	88.22	37.96	25
Triple	2.2	2.79	2
Walk	26.53	15.06	15

Table 4: Predicted vs. Actual Event Counts for Luis Arraez

For Aaron Judge, the transformer model predicts a higher number of home runs (25.60) and doubles (15.66) compared to the XGBoost model's predictions (13.94 home runs and 11.04 doubles). The transformer's predictions for home runs are closer to the actual count of 32, indicating a better adaptation to Judge's power-hitting tendency. Both models predict a similar number of singles, with the transformer slightly higher (42.85) than XGBoost (37.97), compared to the actual count of 36.

Regarding strikeouts, both models perform comparably, with predictions close to the actual count of 107. The transformer model predicts 105.76 strikeouts, while XGBoost predicts 107.50. This suggests that both models effectively capture Judge's strikeout rate.

For Luis Arraez, the transformer model predicts significantly more singles (100.42) than the XGBoost model (64.54), aligning more closely with the actual count of 116. This reflects Arraez's contact-hitting profile. The transformer model also predicts fewer strikeouts (37.96) compared to XGBoost (88.22), approaching the actual count of 25. Although the transformer overestimates strikeouts, it substantially outperforms XGBoost in capturing Arraez's low strikeout tendency.

In terms of home runs, the transformer model predicts fewer (6.27) than XGBoost (12.59), which is closer to the actual count of 9. This suggests the transformer better recognizes Arraez's lower propensity for hitting home runs.

We also analyzed the predicted hit locations to assess how well each model captures the players' tendencies in directing the ball. Tables 5 and 6 present the predicted and actual counts for hit locations.

Hit Location	XGBoost Predicted	Transformer Predicted	Actual Count
Pitcher	5.5	4.75	2
Catcher	2.22	2.08	0
First Base	6.4	8.09	4
Second Base	15.84	15.46	13
Third Base	22.07	27.03	26
Shortstop	24.58	26.63	22
Left Field	24.18	33.42	40
Center Field	28.86	37.83	38
Right Field	27.42	32.63	25

Table 5: Predicted vs. Actual Hit Locations for Aaron Judge

Hit Location	XGBoost Predicted	Transformer Predicted	Actual Count
Pitcher	12.46	22.22	14
Catcher	3.26	3.4	1
First Base	30.67	29.75	31
Second Base	44.17	58.75	56
Third Base	20.41	28.26	23

Shortstop	30.03	48.39	37
Left Field	51.11	82.35	93
Center Field	52.24	65.59	92
Right Field	43.27	37.28	52

Table 6: Predicted vs. Actual Hit Locations for Luis Arraez

For Judge, the transformer model predicts higher counts of hits to the outfield positions—Left Field, Center Field, and Right Field—aligning more closely with the actual counts. For instance, it predicts 33.42 hits to Left Field and 37.83 to Center Field, compared to XGBoost's 24.18 and 28.86, respectively. The actual counts are 40 for Left Field and 38 for Center Field. This suggests the transformer model better captures Judge's tendency to hit deep into the outfield.

For Arraez, the transformer model predicts higher counts of hits to infield positions like Second Base and Shortstop, which corresponds with his contact-hitting style that often results in ground balls and line drives. The transformer predicts 58.75 hits to Second Base and 48.39 to Shortstop, closer to the actual counts of 56 and 37, respectively, than XGBoost's predictions.

Moreover, the transformer's prediction of hits to Left Field (82.35) is closer to the actual count (93) compared to XGBoost's 51.11. This further indicates the transformer's enhanced ability to capture Arraez's hitting patterns.

The transformer-based model demonstrates a superior ability to adapt its predictions to the specific tendencies of individual players. By leveraging sequential and contextual information, it more accurately predicts key statistics for both a power hitter and a contact hitter without explicit player identifiers.

For Aaron Judge, the transformer model better predicts home runs and doubles, reflecting his power-hitting profile. For Luis Arraez, it more accurately forecasts singles and lower strikeout numbers, aligning with his contact-hitting style. The XGBoost model, while capturing general trends, does not adjust its predictions as effectively based on player-specific tendencies.

These findings highlight the transformer's capacity to model nuanced player behaviors, offering valuable insights for personalized strategic decisions in pitching and defensive alignments. The ability to adapt to individual players enhances the model's practical utility in real-world baseball analytics.

4.3 Optimal Pitch Selection Framework & Case Study

To demonstrate the practical utility of our transformer-based model, we developed an optimal pitch selection framework designed to identify the most effective pitch for inducing a desired

outcome in a specific game situation. This framework leverages the model's predictive capabilities to simulate and evaluate various pitch scenarios, ultimately guiding strategic decision-making for pitchers and coaches.

The framework operates by simulating potential pitches and ranking them based on their predicted probabilities of achieving a specified result. For the current batter, we begin by constructing a contextual sequence using the most recent 399 pitches they have faced. This sequence provides the input context for the model, mirroring the data used during training, and captures the batter's recent tendencies and responses to different pitch types and locations.

We then define the desired pitch outcome, specifying both the event (e.g., 'Field Out', 'Strikeout') and, if applicable, the hit location (e.g., 'Shortstop', 'Center Field'). Additionally, we may identify events and hit locations to minimize, such as 'Home Run' or 'Left Field', reflecting strategic considerations to avoid unfavorable outcomes.

For the current pitcher, we generate all feasible combinations of pitch types they are known to throw and possible strike zone locations. (For this study, we use 13 different strike zone locations, Figure 3. However, this approach can be generalized to use any number of strike zone locations, with a higher number enhancing the granularity of the framework's pitch recommendations). Each combination represents a potential pitch that could be delivered in the game situation. To ensure realism, we assign average values for continuous pitch features like velocity and spin rate, calculated based on the pitcher's historical data for each pitch type. This process maintains consistency with the pitcher's typical performance and enhances the credibility of the simulations.

Each simulated pitch is appended as the 400th pitch in the batter's sequence, effectively creating multiple sequences that differ only in the final pitch. The transformer model processes each sequence and outputs probability distributions over possible pitch results and hit locations for the simulated pitch. These probabilities reflect the model's assessment of how likely each outcome is, given the batter's recent history and the characteristics of the simulated pitch.

To rank the simulated pitches, we compute a pitch score for each based on a weighted combination of the probabilities of the desired outcomes and hit locations, while subtracting the probabilities of undesired events or hit locations to penalize pitches likely to result in unfavorable outcomes. Mathematically, the score S for a simulated pitch is calculated as:

$$S = \omega_1 \cdot \sum(P_{DesiredOutcomes}) + \omega_2 \cdot \sum(P_{DesiredHitLocations}) - \omega_3 \cdot \sum(P_{UndesiredOutcomes} + P_{UndesiredHitLocations})$$

Equation 1: Pitch Score

where ω_1 , ω_2 , and ω_3 are adjustable weights reflecting the relative importance of each term, and P denotes the model's predicted probabilities. The weights allow for customization based on strategic priorities, such as emphasizing the desired event over the hit location or placing greater importance on avoiding specific undesired outcomes.

After calculating the scores, we rank the simulated pitches accordingly. The pitch with the highest score is considered optimal, as it maximizes the likelihood of the desired outcome while minimizing the risk of undesired events.

4.3.1 Case Study

To illustrate the practical application and effectiveness of our optimal pitch selection framework, we conduct a case study involving various game scenarios featuring batter Shohei Ohtani and pitchers Gerrit Cole and Gregory Santos. This case study demonstrates the model's ability to adapt to specific situations and players, providing data-driven recommendations that align with strategic objectives.

Scenario 1: First Pitch of the Game

In the first scenario, we consider the initial pitch of the game, where the pitcher aims to gain an early advantage by getting ahead in the count. The strategic objective is to maximize the probability of a strike while minimizing the likelihood of the batter achieving a hit. To formalize this within our framework, we define the desired event as 'Strike' and the undesired events as all 'Hit' events (i.e., 'Single', 'Double', 'Triple', 'Home Run'). The score S , as defined earlier (Equation 1), is calculated for each simulated pitch using the specified weights: 0.4 for the desired event, 0.6 for minimizing undesired events, and 0 for hit location. The score reflects the trade-off between increasing the likelihood of a strike and reducing the risk of a hit, with no emphasis placed on the predicted hit location in this scenario.

Applying the framework with these parameters, the model simulates all feasible pitch types and zones for Gerrit Cole, incorporating his pitch repertoire and historical performance data. The top three pitches recommended by the model are knuckle curveballs delivered to zones 2, 5, and 3, respectively. These zones correspond to specific areas within the strike zone from the perspective of the Catcher, as illustrated in Figure 3. The recommendation aligns with strategic pitching practices, as knuckle curveballs can be effective in inducing initial strikes against batters who may be anticipating fastballs early in the count.

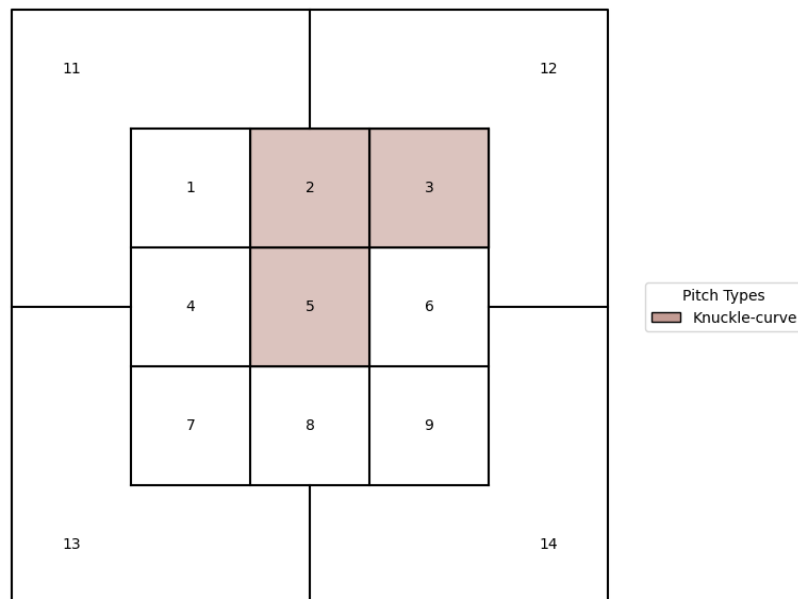


Figure 3: First Strike Pitch Recommendations

Scenario 2: Advancing the Count to 0-2

Building on the first scenario, we assume that the first recommended knuckle curveball to zone 2 resulted in a strike, followed by a fastball to zone 2, also resulting in a strike, bringing the count to 0-2. At this juncture, the pitcher aims to capitalize on the advantageous count by attempting to strike out the batter while minimizing the risk of a hit. The desired event is now set to 'Strikeout', with the undesired events remaining as all 'Hit' events. The weights are adjusted accordingly to reflect the increased emphasis on achieving a strikeout.

Under these conditions, the model recommends low fastballs delivered to zones 7, 8, and 9 (Figure 4). These zones are located at the bottom of the strike zone, aligning with Shohei Ohtani's known tendency to strikeout on pitches lower in the zone (Appendix A). This recommendation is consistent with Gerrit Cole's strengths, as his fastball is among his most effective pitches (Appendix B). By targeting Ohtani's weakness against low pitches, the model's suggested approach capitalizes on both the batter's tendencies and the pitcher's strengths, maximizing the chance of a strikeout.

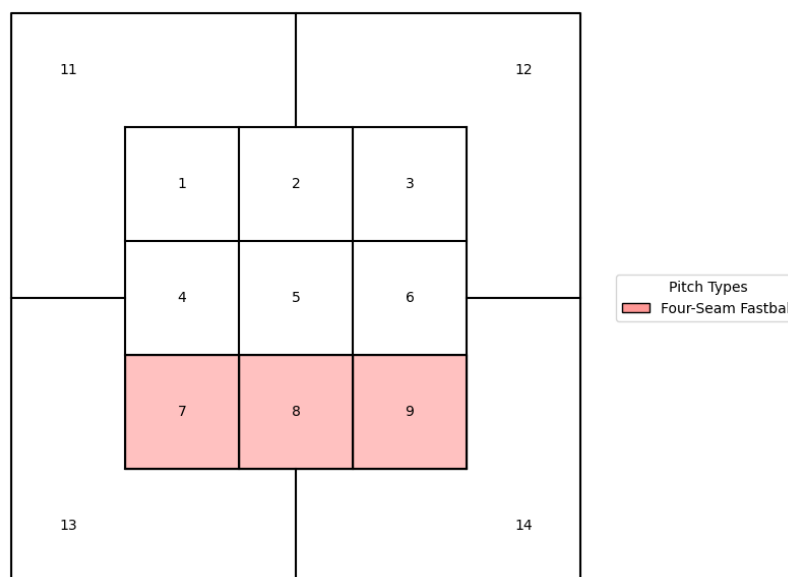


Figure 4: Gerrit Cole Strikeout Pitches

To further assess the model's adaptability, we replace Gerrit Cole with a different pitcher, Gregory Santos, known for his breaking pitches rather than a dominant fastball (Appendix C). In the same 0-2 count scenario, the model now recommends a slider to zones 1 and 4, or a sinker to zone 9 (Figure 5). This shift in recommendations reflects the model's consideration of the pitcher's strengths and historical performance. By suggesting pitches that align with Santos's repertoire, the model demonstrates its capacity to tailor strategies based on the specific pitcher, enhancing the effectiveness of the pitch selection.

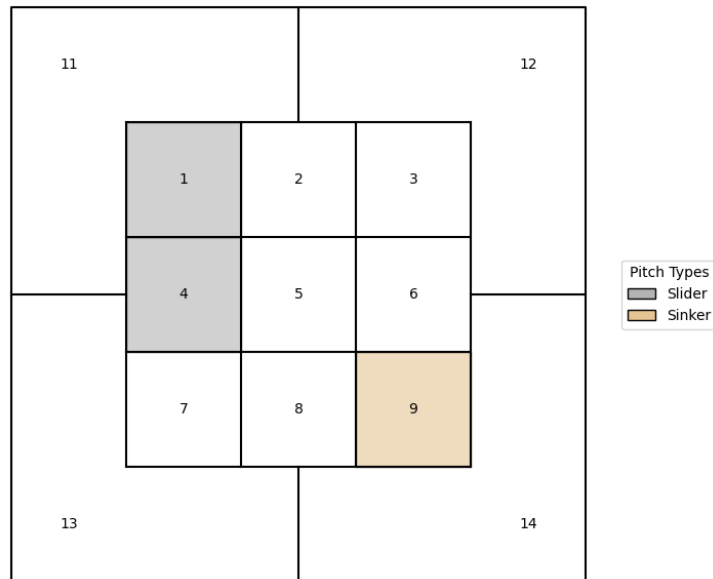


Figure 5: Gregory Santos Strikeout Pitches

Scenario 3: Inducing a Ground Ball for a Double Play

In the final scenario, we reset the count and return to Gerrit Cole as the pitcher, with the game context now including a runner on first base. The strategic objective is to induce a ground ball that could result in a double play, thereby minimizing the opponent's scoring opportunity. To achieve this, we set the desired event to 'Field Out' and the desired hit locations to infield positions, specifically targeting hit locations associated with ground balls to infielders. The undesired events remain as all 'Hit' events to reduce the chance of the batter reaching base safely. The weights are assigned as 0.4 for the desired event, 0.3 for the desired hit location, and 0.3 for minimizing undesired events, balancing the importance of inducing an out, directing the ball to the infield, and avoiding hits.

Applying the framework with these parameters, the model recommends cutters delivered to zones 3 and 12, or a changeup to zone 4 (Figure 6). The high and inside cutters are likely intended to jam the batter, leveraging their speed and horizontal movement to induce weak contact. The recommendation of the changeup to zone 4 may reflect its potential to induce a ground ball in this specific scenario, possibly based on patterns observed in the batter's recent performance captured by the context window. Figure 7 presents a heatmap of the model's predicted hit locations for these recommended pitches, illustrating the concentration of likely contact toward the infield positions.

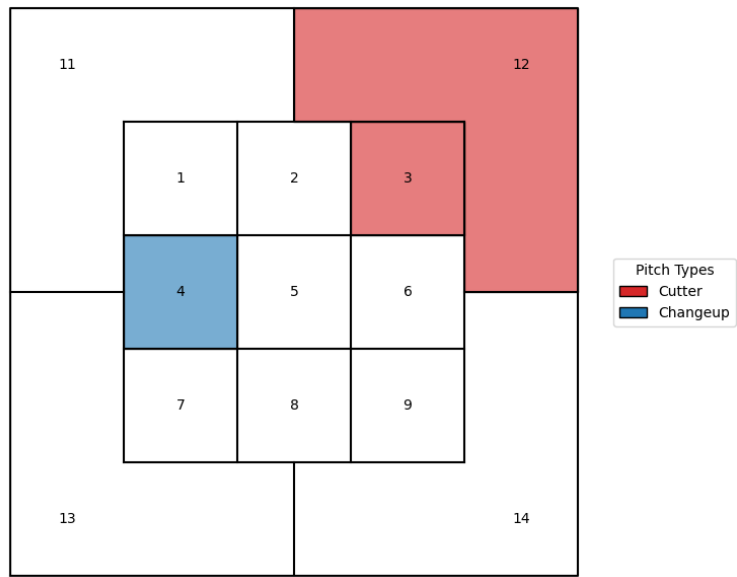


Figure 6: Ground Ball Inducing Pitches

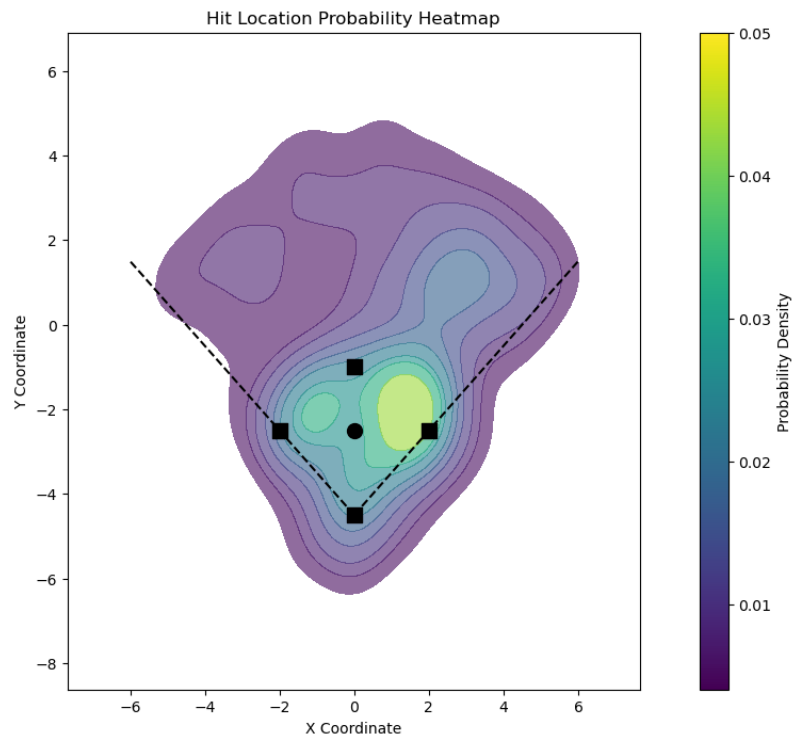


Figure 7: Hit Location Heatmap for Ground-Ball Inducing Pitches

These scenarios highlight the model's ability to adapt its recommendations based on the specific game context, batter-pitcher matchups, and strategic objectives. By simulating potential pitches and evaluating their predicted outcomes, the framework provides actionable insights that align with both the pitcher's strengths and the desired tactical outcomes. The model's recommendations consider not only the probabilities of achieving the desired events but also the minimization of undesired outcomes, offering a balanced approach to risk and reward.

5. Conclusion

In this study, we developed a transformer-based neural network model to predict individual pitch outcomes and hit locations in Major League Baseball. By leveraging sequences of pitches and rich contextual information, our model outperformed baseline models, including a historical average and an XGBoost classifier. The transformer model excelled in predicting less frequent but strategically significant events such as singles, doubles, and home runs, and demonstrated enhanced adaptability to individual player tendencies without relying on explicit player identifiers.

The optimal pitch selection framework further showcased the model's practical utility by providing actionable insights tailored to specific game contexts, batter profiles, and pitcher strengths. The case study involving Shohei Ohtani and Gerrit Cole illustrated how the model can inform strategic decisions, optimize pitch selection, and enhance defensive alignments. By predicting both pitch outcomes and hit locations, our model bridges the gap between advanced analytics and real-time decision-making, offering comprehensive support for coaches and players.

5.1 Future Work

While our model has demonstrated significant capabilities, several avenues exist for further enhancement.

Our current approach focuses primarily on the batter's historical pitch sequence, effectively being "playerless" from the pitcher's perspective. In practice, both pitchers and batters adjust their strategies based on knowledge of each other's tendencies. Incorporating explicit information about the current pitcher could enhance the model's predictive accuracy and adaptability. One potential direction is to create pitcher embeddings analogous to the batter embeddings generated from the context window. By using sequences of the pitcher's previous pitches, the model could learn representations that capture the pitcher's style, strengths, and tendencies. Including detailed information about the pitcher's arsenal—such as pitch types, average velocities, and usage frequencies—could provide valuable context, leading to more nuanced predictions and strategic recommendations.

In the current model, certain events are grouped into broader categories to streamline the classification task (e.g., grouping "Fielder's Choice," "Fly Ball," and "Field Out" under "Field Out"). While this simplifies the model and reduces sparsity, it abstracts away granular details that could be valuable for strategic planning. Future work could explore separating these grouped events into more specific categories, increasing the granularity of predictions. By distinguishing between

different types of outs or hits, the model might provide more detailed insights into likely outcomes, enhancing defensive strategies and player positioning. Addressing potential challenges related to class imbalance and data sparsity would be essential to maintain high predictive performance with an expanded classification task.

By focusing on these areas, we aim to further refine the model's capabilities and increase its applicability in real-world baseball analytics. Incorporating pitcher context and increasing prediction granularity could enhance the model's precision and strategic value, contributing to the evolving landscape of baseball analytics and decision-making.

6. References

- [1] Heaton, C., & Mitra, P. (2021). Using Machine Learning to Describe How Players Impact the Game in the MLB. In The 16th Annual MIT Sloan Sports Analytics Conference.
- [2] Heaton, C., & Mitra, P. (2023). Using Machine Learning to Describe How Players Impact the Game in the MLB. Learning Contextual Event Embeddings to Predict Player Performance in the MLB. In The 17th Annual MIT Sloan Sports Analytics Conference.
- [3] Melville, W., Melville, J., Dawson, T., Nieves-Rivera, D., Archibald, C., & Grimsman, D. (2023). A Game Theoretical Approach to Optimal Pitch Sequencing. In The 17th Annual MIT Sloan Sports Analytics Conference.
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [5] Statcast Database. Retrieved from Baseball Savant: <https://baseballsavant.mlb.com>.
- [6] LeDoux, J. (2017). pybaseball: A Python Package for Baseball Data Analysis. Retrieved from <https://github.com/jldbc/pybaseball>.
- [7] TangoTiger (2020). Run Values by Pitch Count. Retrieved from: <http://tangotiger.com/index.php/site/article/run-values-by-pitch-count>
- [8] Major League Baseball Advanced Media. (2024). Baseball Savant. Retrieved from <https://baseballsavant.mlb.com/>
- [9] Silver, J., & Huffman, T. (2021). Baseball Predictions and Strategies Using Explainable AI. In The 16th Annual MIT Sloan Sports Analytics Conference.
- [10] Clemens, B. (2022). Can "hard in and soft away" make your troubles go away? In FanGraphs Blog. Retrieved from <https://blogs.fangraphs.com/can-hard-in-and-soft-away-make-your-troubles-go-away/>
- [11] Crawshaw, M. (2020). Multi-Task Learning with Deep Neural Networks: A Survey. Retrieved from <https://arxiv.org/abs/2009.09796>.

[12] PyTorch Documentation. CrossEntropyLoss. Retrieved from <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>.

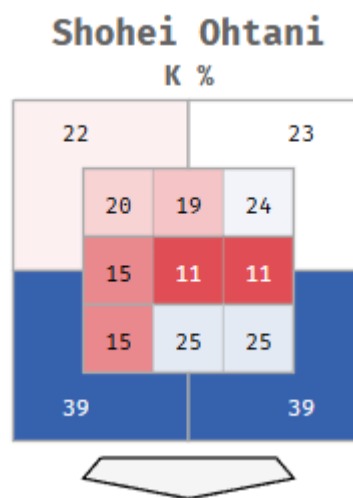
[13] PyTorch Documentation. MeanSquaredErrorLoss. Retrieved from <https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html>.

[14] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM.

Appendix

A. Shohei Ohtani Strikeout Rate by Zone 2023

The below figure shows Shohei Ohtani's strikeout rates by zone for the 2023 MLB season. The number in each zone represents the percentage of Ohtani's total strikeouts in 2023 that occurred on a pitch to that zone [8].



B. Gerrit Cole Run Value by Pitch Type 2023

The below table shows Gerrit Cole's run value and run value per 100 pitches by pitch type for the 2023 MLB season. Run value is the impact of an event based on the runners on base, outs, ball and strike count, with a higher number indicating higher value [7, 8].

Pitch Type	Run Value	Run Value/100
------------	-----------	---------------

4-Seam Fastball	29	1.7
Slider	9	1.4
Curveball	4	1.0
Changeup	2	0.7
Cutter	0	0.2

C. Gregory Santos Run Value by Pitch Type 2023

The below table shows Gregory Santos' run value and run value per 100 pitches by pitch type for the 2023 MLB season. Run value is the impact of an event based on the runners on base, outs, ball and strike count, with a higher number indicating higher value [7, 8].

Pitch Type	Run Value	Run Value/100
Slider	17	3.1
4-Seam Fastball	1	3.6
Changeup	0	-2.8
Sinker	-8	-1.8