



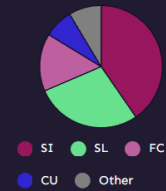
Objective

- Design a predictive model to provide pitch sequence recommendations
- Analyze Pitcher and Batter tendencies
- Provide a probability distribution across pitcher's available pitch types
- Create a Web App for players/coaches to interact with the model

Solution

- Developed a Recurrent Neural Network
- Predict pitch group, then specific pitch types
- Evaluate the feasibility of each pitch using several sub models

Pitch 1: SI (Sinker)
Count: 1-0



Technical Challenges

1. **Fastball Class Imbalance** – Fastballs are the most common pitch in baseball, which causes the model to favor recommending fastballs
2. **Limited Batter Features** – Only had access to public data to train the RNN, no swing metrics
3. **Model Design** – One RNN vs. a Two Stage Approach, predicting pitch group then type.

Evaluate Outs

An XGBoost classifier predicts the likelihood of each out type for a given pitch and matchup. The model uses this to steer recommendations toward the optimal out type for that at-bat.

Evaluate State

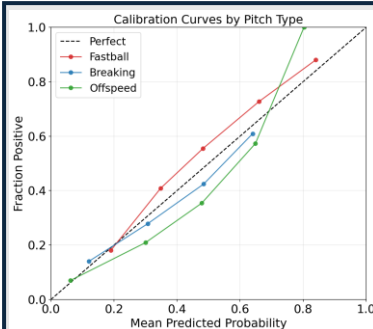
Another XGBoost classifier predicts whether a pitch will result in a ball or strike, to help transition the model to the next game state. The count is often the driving factor of pitch selection.

Evaluate RE

Our model uses a precomputed Run Expectancy (RE) table to steer recommendations toward favorable pitcher results. RE is the expected runs scored from a game state to inning's end.

Results

The Pitch Group RNN, which classifies each pitch as a fastball, breaking ball, or off-speed pitch, achieves 53% accuracy. Despite this, the model reaches 88% top-2 accuracy, meaning the correct pitch group appears among the model's top 2 predictions 88% of the time. The Calibration Curves below show how well the model's confidence aligns with its actual correctness for each pitch group. The model tends to be underconfident on fastballs and overconfident on breaking and off-speed pitches. This reflects the class imbalance of our training data, which we plan to address further in the future.



Pitch Ranking

A power mean formula combines the RNN, out type, and run expectancy scores into a single value, where the exponent p controls how much weight is given to the strongest-performing component (a higher p -value amplifies the influence of the larger values of the input).

$$M_p(a_1, a_2, \dots, a_n) = \left(\frac{1}{n} \sum_{k=1}^n a_k^p \right)^{1/p}$$

Model Architecture

This diagram highlights the 5 stages of our pitch inference process.

