Modeling Pitch Velocity Differentials: Thought Process and Rationale

By Kaleb Jordan

In the modern era of baseball analytics, understanding how pitchers manipulate pitch speeds and movement to deceive hitters has become an essential area of study. This project was motivated by a practical challenge: building a predictive model that estimates the **velocity differential** between a pitcher's fastball and breaking pitches based on their mechanics, pitch characteristics, and usage patterns. Beyond building a predictive model, I also wanted to construct an interactive application that allows users to explore predictions, residuals, and pitch-level patterns for individual pitchers across seasons.

Data Cleaning and Preparation

The original dataset contained over 3 million pitches spanning the 2020 to 2024 MLB seasons. Variables included pitch velocity, spin, break measurements, release and location coordinates, and more. The dataset also contained significant amounts of missing data, especially in variables related to batted ball outcomes. Since the focus of the project was on **pitch mechanics** rather than what happened after the pitch was thrown, I chose to remove outcome variables (i.e., exit velocity and launch angle), which were both irrelevant and contained excessive missing values.

For critical mechanical variables — **release velocity**, **extension**, **break_x**, **break_z**, and **spin measurements** — I carefully assessed the level of missingness. Missing data in these variables was relatively low (under 1% in most cases), so I opted for **listwise deletion** where feasible. However, to showcase alternative imputing methods, since variables, **spin rate** and **spin direction**, had slightly higher levels of missingness and to preserve the dataset's richness and prevent losing thousands of rows, I applied **regression-based imputation** for these two features. I built season-specific linear models leveraging correlated variables like break, release metrics, and pitch velocity. This approach maintained consistency while allowing for subtle shifts in relationships across seasons.

I also created a **clean pitcher-season-pitch type aggregation**, keeping only pitcher-seasons with a sufficient number of total pitches (above the 25th percentile or 165 pitches). This filtering ensured that outlier pitcher-seasons with very few pitches did not distort model training or artificially inflate variance estimates.

Feature Engineering

Understanding that **velocity gaps** are not just a function of raw velocity but also influenced by **pitcher mechanics**, **release consistency**, and **pitch variability**, I engineered a set of features that captured both central tendencies and ranges of variability:

- For each pitcher-season and pitch type, I calculated **means** for spin rate, spin direction, break_x, break_z, release locations, extension, and vertical/horizontal approach angles.
- I also calculated **quantile ranges** (75th - 25th percentile) to represent the variability in these mechanics.
- For usage patterns, I computed **fastball usage** and **breaking ball usage** percentages.

I included a **reliever flag**, indicating whether a pitcher always appeared after the 5th inning (reliever). This variable provided insight into how role and usage patterns might relate to pitch velocity differentials.

Finally, I ensured categorical variables like **pitch_type** and **throws** (Throwing hand) were properly encoded as factors. This allowed for **interaction terms** in modeling, which proved important for capturing pitch-type-specific effects.

Model Creation and Validation

1. Linear Regression Model

My primary model was a **linear regression** predicting the velocity gap between fastballs and breaking balls using the mechanical and usage features described above. I included key interaction terms, such as **pitch_type × mean_induced_break_z**, to account for how different pitch types may respond differently to changes in break characteristics.

Before finalizing the model, I assessed:

- **Multicollinearity** using VIF (Variance Inflation Factor), ensuring no predictors (other than intentionally constructed interactions) exceeded acceptable thresholds.
- **Residuals** for normality, homoscedasticity, and the presence of outliers.
- **Cross-validation** results to confirm generalizability.

The linear model achieved:

- **Training RMSE**: 0.407 mph
- **Testing RMSE**: 0.428 mph
- **Mean Cross-validation RMSE:** 0.409 mph
- **Adjusted R^2**: 0.99

The consistency between training, testing, and cross-validation RMSE values demonstrated strong model stability and low overfitting risk. I believe the model generalizes well and is a valid model.

2. Random Forest and Gradient Boosting Models

To benchmark against more complex algorithms, I also trained:

- A **Random Forest** model, which achieved low training RMSE (0.346) but suffered from overfitting, with test RMSE rising to 0.803.
- A **Gradient Boosting Machine (GBM)** model, which balanced bias and variance better, with test RMSE around 0.664. The difference between the training and test set was still high and therefore still suffers from overfitting.

While both models provided valuable insights into variable importance, neither generalized as well as the linear model. This outcome suggested that the relationship between the features and the velocity gap was largely **linear or linear with modest interactions**, making linear regression the most appropriate choice for this problem.

Residual Analysis and Insights

I examined the **largest residuals** to identify where the model struggled most. Many outliers were pitchers known for unconventional pitch profiles or deliberate variation in velocity (for example, Tyler Rogers and Alec Mills). This finding validated the model's accuracy while also highlighting the natural limits of any mechanical model in accounting for strategic deception or inconsistency.

I also conducted exploratory visualizations comparing predicted and actual gaps across pitch types and seasons. These confirmed that while the model generalized well, certain pitcher-specific effects remained — an expected and valuable finding that points toward potential future model enhancements.

Interactive Application Development

To make the model results accessible and actionable, I developed a **Shiny web application**. The app allows users to:

- Filter results by **pitcher**, **season**, and **pitch type**.
- View **residuals tables** for selected subsets.
- Visualize **predicted vs. actual velocity gaps** interactively.

Why a Shiny app?

- Static tables and plots cannot capture the dynamic nature of pitch analysis.
- Coaches, analysts, and data scientists need to explore data subsets in real time, which the app facilitates.

The app not only serves as a visualization tool but also provides an **error diagnostic interface**, helping identify pitchers and pitch types where model predictions deviate from reality — valuable for player development and scouting.

Conclusions and Future Directions

This project demonstrated the value of combining **domain expertise**, **statistical modeling**, and **interactive tools** to analyze a complex baseball performance metric. The process highlighted that with thoughtful feature engineering and validation, simple interpretable models can match or even outperform black-box machine learning methods when applied to structured sports data.

Moving forward, potential extensions include:

- Incorporating **batter matchup data** to understand how velocity gaps interact with opponent tendencies.
- Adding **game context features** (count, score leverage) to capture situational adjustments.
- Testing **time-series models** to examine how pitcher mechanics and velocity gaps evolve during a season.

By balancing statistical rigor with practical applicability, this project offers both immediate insights and a flexible platform for ongoing research in baseball analytics.