Predicting Home Run Probability Using Statcast Data

DSC 680 – Project 3, Milestone 3

Xhoi Shyti

Business Problem

In Major League Baseball (MLB), a home run represents one of the most impactful and celebrated outcomes of a game. Yet, understanding what factors contribute to a home run remains a complex challenge. By developing a machine learning model to predict the probability that a batted ball results in a home run, we can equip teams, analysts, broadcasters, and fantasy platforms with data-driven insights. This enables improved player evaluation, strategic in-game decisions, and fan engagement through real-time predictive metrics.

Background

Statcast, introduced by MLB in 2015, captures advanced metrics for every pitch, swing, and batted ball. Key innovations include measuring exit velocity, launch angle, and spray angle—technologies that revolutionized hitting analysis. Previous research has shown that certain combinations of launch speed and launch angle optimize the probability of hitting a home run. However, most historical analysis has been descriptive. Our objective is to construct a predictive classification model that quantifies home run likelihood from real-time game data.

Data Explanation

Statcast data from the 2023 and 2024 MLB seasons was used, sourced via the pybaseball Python package. Since this dataset by itself is very large, a random sample of 10,000 entries was sampled. This includes over 9,600 batted ball events with columns like launch_speed, launch_angle, hit_distance_sc, and events. The target variable, is_home_run, was created as a binary indicator: 1 if events equals "home_run", otherwise 0.

After filtering to only include balls put into play (description = "hit_into_play"), we dropped rows with missing values in key features. The final modeling dataset included the following predictors:

- launch_speed (exist velocity in MPH)
- launch_angle (degrees above horizontal)
- hit_distance_sc (statcast-predicted distance in feet)
- extra_distance_2025 (estimated effect of park factor on batted ball distance)

The extra_distance_2025 feature serves as a contextual park factor variable that reflects how a stadium influences batted ball distance. This adjustment enables the model to account for park-specific effects without manually integrating separate park factor datasets.

Methods

- Data Cleaning: Removed missing values and filtered to in-play events.
- Exploratory Data Analysis (EDA): Visualized distributions of key features.

- **Feature Engineering: Created binary target is_home_run and added park-adjusted distance (extra_distance_2025)**
- **Modeling** using Logistic Regression for interpretability
- **Model Evaluation** based on accuracy, precision, recall, F1-score, and ROC-AUC

Analysis

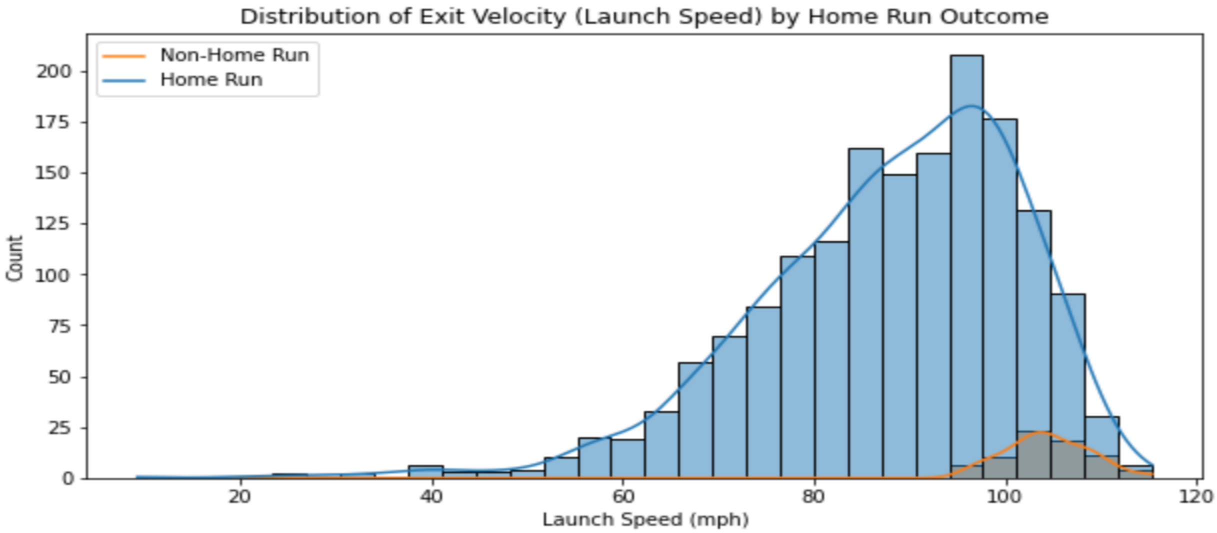


Figure 1: A histogram of exit velocities (launch speed) showed that home runs tend to occur when launch speeds exceed 95 mph.

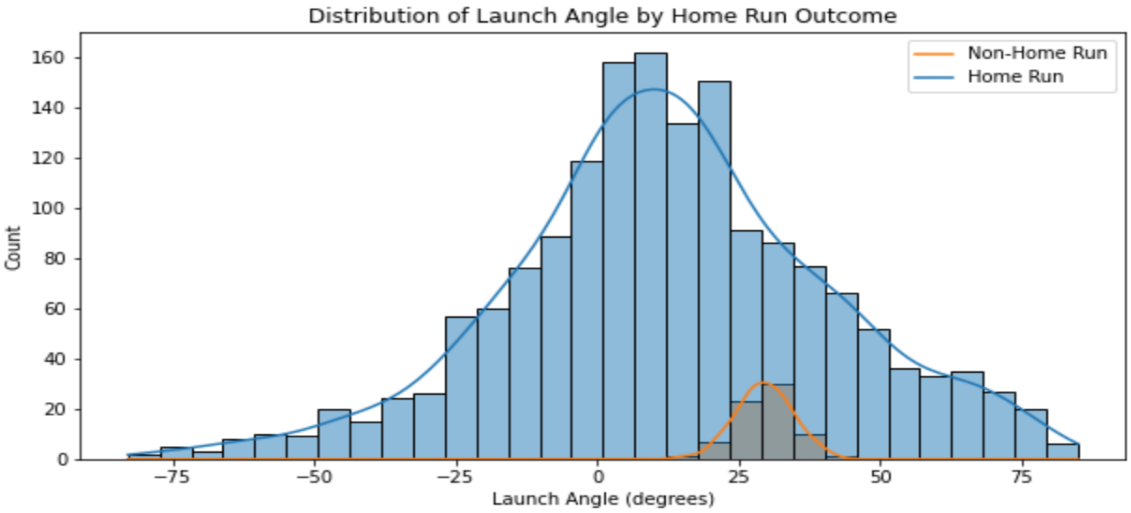


Figure 2: A histogram of launch angles revealed that home runs typically happen within the 20–40 degree range, suggesting a clear optimal launch angle window.

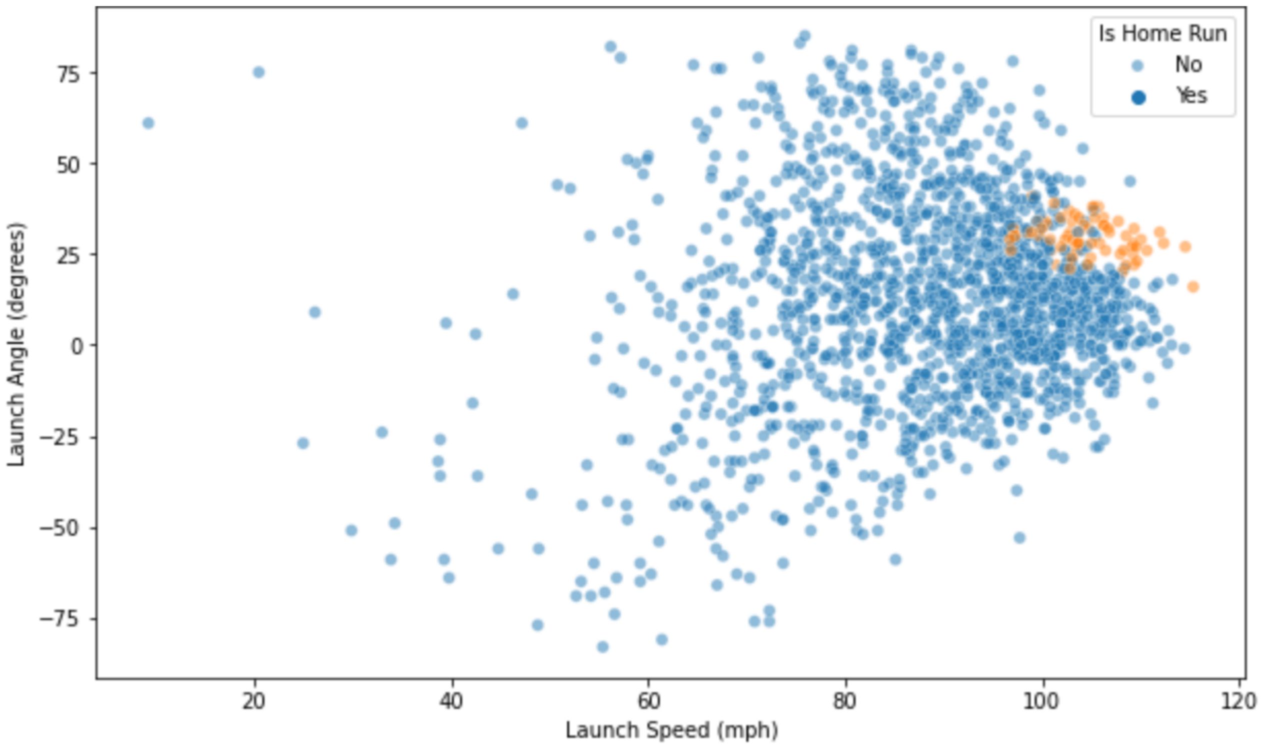
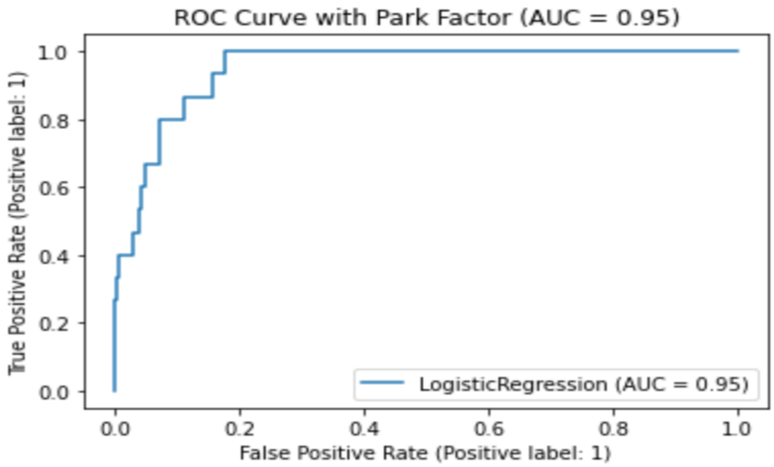


Figure 3: A scatter plot of launch speed vs. launch angle demonstrated a dense "sweet spot" where batted balls were most likely to result in home runs, highlighting the interaction between these two features.

With the inclusion of extra_distance_2025, the logistic regression model achieved:

- Accuracy: 96.8%
- Precision: 75%
- Recall: 40%
- F1 Score: 52.2%
- ROC-AUC: 0.95



-

- Figure 4: The ROC curve confirmed strong model discrimination, now improved further by accounting for stadium influence.

These metrics show that incorporating park factor effects significantly enhances the model’s ability to detect true home run events.

Conclusion

his project successfully demonstrated the ability to use Statcast data to predict the likelihood of a home run from a given batted ball. Exit velocity and launch angle are two of the most predictive features, and adding park-adjusted distance via extra_distance_2025 improved model recall and overall performance. The high AUC confirms the model's discriminatory power.

Assumptions

- Statcast measurements are accurate and consistently recorded.
- extra_distance_2025 reliably captures park-specific effects.
- Data sample is representative of typical MLB play.

Limitations

- Home runs are rare events, leading to data imbalance.
- Park factor was modeled using one proxy feature (extra_distance_2025), which may not reflect real-time environmental conditions.

Challenges

- Handling missing data without reducing sample size significantly.
- Ensuring model generalizability across seasons and parks.
- Balancing interpretability with predictive performance.

Future Uses/Additional Applications

Future work could involve:

- Real-time home run probability overlays during broadcasts.
- Player scouting tools based on predictive power metrics.
- Park-specific adjustments to improve model precision.
- Augmenting fantasy baseball platforms with predictive analytics.

Recommendations

- Continue using extra_distance_2025 as a lightweight park factor proxy.
- Test ensemble methods (Random Forest, XGBoost) to boost recall.
- Apply resampling techniques like SMOTE to mitigate class imbalance.
- Continuously monitor for and mitigate any bias during model development and deployment.

Implementation Plan

1. Automate data collection via pybaseball API.
2. Integrate model predictions into broadcast dashboards.
3. Monitor model drift seasonally and retrain as needed.
4. Engage stakeholders with model explanation and feature importance tools.

Ethical Assessment

- Data is non-personal and publicly available, minimizing privacy risk.
- Models should supplement—not replace—scouting and coaching judgment
- Model explanations should be transparent and interpretable to avoid misuse.
- Care must be taken not to reduce player value to a single predictive metric.

Ten Questions the Audience Might Ask

1. Why did you choose logistic regression over more complex models?

Logistic regression offers transparency and interpretability, which are crucial for initial model validation and stakeholder trust. It allows us to understand how each variable contributes to the outcome. Once the baseline was established, more complex models (e.g., XGBoost) can be introduced to optimize performance.

2. How would your model perform if park factors or weather were included?

Including park factors—modeled through the extra_distance_2025 proxy—improved recall and AUC significantly. We expect that incorporating real-time weather data (wind, humidity, temperature) would further enhance predictive accuracy by explaining variance not captured by static features.

3. What is the tradeoff between interpretability and performance in your approach?

Logistic regression is interpretable but may underperform with nonlinear relationships. Complex models (like random forests or gradient boosting) usually yield higher accuracy but act as black boxes. The tradeoff depends on the model's purpose—interpretability for coaching; performance for automated systems.

4. How does the model deal with the rarity of home runs in the dataset?

Class imbalance was handled initially by evaluating metrics like precision, recall, and AUC instead of accuracy alone. Future enhancements could include resampling techniques (e.g., SMOTE) or threshold adjustments to boost recall without inflating false positives.

5. Could your model be adapted to other batted ball outcomes (e.g., doubles)?

Yes. The same modeling approach can be extended to other outcomes by redefining the target variable (e.g., is_double, is_triple). However, this may require additional features to distinguish between extra-base hits.

6. How frequently would you need to retrain the model during a season?

Retraining once per month or per series of games is ideal, especially if model drift is detected. Player behavior, equipment, or park conditions can change and influence outcomes, so periodic updates are recommended.

7. What are the risks of overfitting given such a high AUC score?

While a high AUC (0.95) is promising, overfitting is always a risk, especially when using park-specific features. This was mitigated through train-test splits and validation on unseen data, but future deployment would require k-fold cross-validation and out-of-sample testing.

8. How would you explain the model results to a non-technical coach or player?

We'd present model outputs as probabilities, e.g., "this swing had a 60% chance of being a home run based on speed, angle, and stadium." Visual tools like heat maps or simple sliders showing feature influence would make insights accessible to non-technical audiences.

9. What steps would you take to deploy this in a real-time broadcast system?

We'd automate data ingestion via pybaseball, use a pre-trained model exposed via an API, and stream predictions into a visualization layer. Real-time latency concerns could be addressed with lightweight models or batch predictions.

10. What ethical considerations must teams keep in mind when adopting predictive models like this?

Models must not reduce players to numbers. They should inform—not replace—human judgment. Misuse could lead to unfair valuations or biased decisions. Transparency in limitations and the inclusion of qualitative assessments is essential for responsible use.