STT 810 Project:

Predicting CPBL Hitter Performance

Wei-Chieh, Tseng: 181608938 Draco Hong: 181591665

Cheng-Lun Lee: 181487980

December 9, 2025

Abstract

This project studies which offensive statistics matter most for hitters in the Chinese Professional Baseball League (CPBL). We use OPS+ as our main measure of hitting performance and combine two modeling approaches. First, we treat OPS+ as a continuous response and fit robust regression (Huber regression), with and without principal component analysis (PCA), to handle outliers and multicollinearity. Second, we define a binary "elite" indicator based on OPS+ and fit logistic regression models using plate discipline and batted-ball statistics such as strikeout rate, BABIP, and balls-in-play rate. With four process-based predictors (PA, K_pct, BABIP, BIP_pct), the logistic model reaches about 87% test accuracy and a test AUC around 0.93 on a hold-out set, and about 89% accuracy on the full sample, suggesting that low strikeout rate and high BABIP are strong markers of elite CPBL hitters. The robust regression results are consistent with this picture and also highlight slugging percentage (SLG) as a key driver of OPS+.

Contents

1 Introduction

OPS+ is a standard sabermetric measure that summarizes a hitter's on-base and slugging performance, adjusted for league and park factors. In this project, we focus on batters from the Chinese Professional Baseball League (CPBL) and ask two related questions:

1. Which offensive statistics are most strongly associated with OPS+?
2. Can we build reasonably accurate models that predict or classify hitter performance using process-based statistics such as plate discipline and batted-ball metrics?

To address these questions, we use CPBL data from the 2024 and 2025 seasons and apply two complementary approaches. First, we treat OPS+ as a continuous outcome and fit robust regression models (Huber regression), with and without dimensionality reduction via principal component analysis (PCA). Second, we convert OPS+ to a binary "elite" indicator and fit logistic regression models based on plate discipline and batted-ball variables.

Our code and cleaned data are available in a public GitHub repository:

https://github.com/neil7227/STT810-project/tree/main

2 Data and Preprocessing

2.1 Data source

We use CPBL batter data from the 2024 and 2025 seasons, obtained from a public baseball statistics website (Robas, *The Baseball Revolution*). The raw dataset contains basic batting statistics for 307 players and 18 columns, including:

- Plate appearances (PA), at-bats, hits, doubles, triples, home runs.
- Walks, strikeouts, hit-by-pitch, sacrifice flies.
- Summary rate statistics such as batting average (AVG), on-base percentage (OBP), slugging percentage (SLG), OPS, and OPS+.

2.2 Cleaning and feature construction

We performed the following preprocessing steps (also documented in our notebooks and cleaned CSV files in the GitHub repo):

1. Removed duplicate or redundant columns such as OPS (since OPS+ is already included) and other highly collinear summary stats.

2. Ensured that numeric variables were stored as numeric types and handled obvious missing or inconsistent entries.

3. Excluded players with very few plate appearances to avoid noise from extremely small samples. For the logistic regression analysis we kept hitters with at least 50 PA, yielding roughly 100 players.

4. Constructed process-based variables, for example:

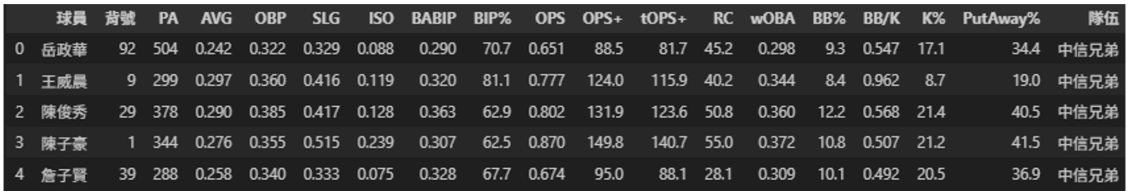
   - Walk rate: BB_pct $= \frac{\text{BB}}{\text{PA}}$.
   - Strikeout rate: K_pct $= \frac{\text{SO}}{\text{PA}}$.
   - Balls-in-play rate: BIP_pct.
   - Batting average on balls in play (BABIP).
   - Put-away rate in two-strike counts (PutAway_pct).

5. Kept OPS+ as the primary performance outcome.

Table 1: Key variables used in the analysis.

Variable	Description
OPS_plus	League- and park-adjusted OPS index (100 = league average)
PA	Plate appearances
BB_pct	Walk rate (BB / PA)
K_pct	Strikeout rate (SO / PA)
BABIP	Batting average on balls in play
BIP_pct	Batted-ball-in-play rate
PutAway_pct	Two-strike put-away rate
SLG	Slugging percentage

2.3 Exploratory analysis



	球員	背號	PA	AVG	OBP	SLG	ISO	BABIP	BIP%	OPS	OPS+	tOPS+	RC	wOBA	BB%	BB/K	K%	PutAway%	隊伍
0	岳政華	92	504	0.242	0.322	0.329	0.088	0.290	70.7	0.651	88.5	81.7	45.2	0.298	9.3	0.547	17.1	34.4	中信兄弟
1	王威晨	9	299	0.297	0.360	0.416	0.119	0.320	81.1	0.777	124.0	115.9	40.2	0.344	8.4	0.962	8.7	19.0	中信兄弟
2	陳俊秀	29	378	0.290	0.385	0.417	0.128	0.363	62.9	0.802	131.9	123.6	50.8	0.360	12.2	0.568	21.4	40.5	中信兄弟
3	陳子豪	1	344	0.276	0.355	0.515	0.239	0.307	62.5	0.870	149.8	140.7	55.0	0.372	10.8	0.507	21.2	41.5	中信兄弟
4	詹子賢	39	288	0.258	0.340	0.333	0.075	0.328	67.7	0.674	95.0	88.1	28.1	0.309	10.1	0.492	20.5	36.9	中信兄弟

Figure 1: Head of the CPBL batter dataset (example rows).

We then restricted to numeric columns and computed the correlation matrix. From the logistic-regression notebook we obtain the following correlation heatmap:

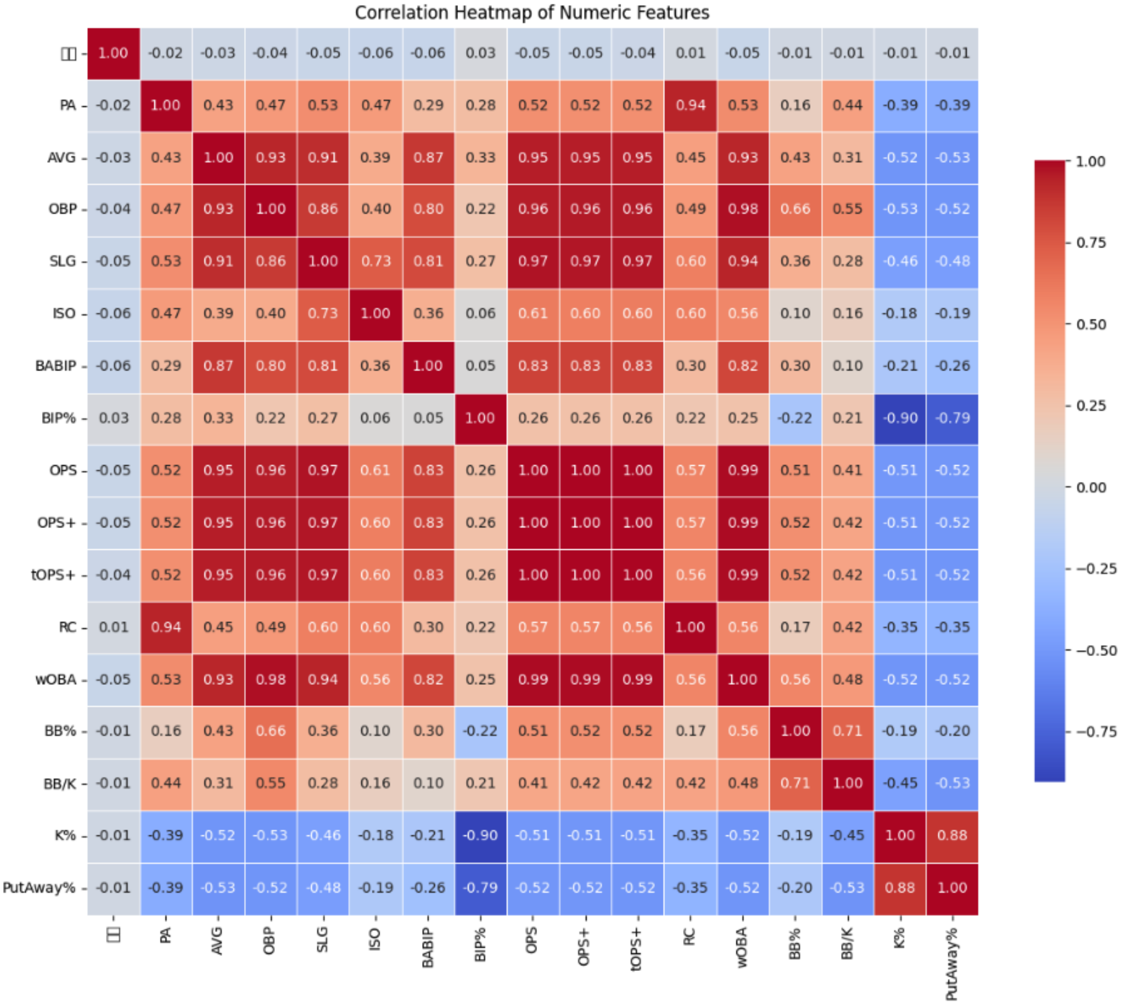


Figure 2: Correlation heatmap of numeric features in the CPBL dataset.

The heatmap shows that slugging percentage (SLG) and several batted-ball variables are strongly associated with OPS+, which motivates their use in the robust regression model. For the logistic regression model, we focus on process-based plate discipline and batted-ball rates instead of directly using OPS+ or SLG as predictors.

3 Methods

3.1 Robust regression for continuous OPS+

3.1.1 Model choice

Baseball data often contain outliers, for example players with very few plate appearances or unusual one-year spikes. Ordinary least squares (OLS) regression can be sensitive to these extreme observations, so we use Huber regression to predict OPS+ from a set of offensive statistics and we compare models based on the original features and on PCA components.

Let y_i denote the OPS+ of player i and $\mathbf{x}_i$ a vector of predictors (e.g., OBP, SLG, BB_pct, K_pct). The Huber regression solves

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} \rho_\delta\big(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta}\big),$$

where $\rho_\delta(\cdot)$ is the Huber loss with tuning parameter δ: it behaves like squared loss for small residuals and absolute loss for large residuals.

3.1.2 PCA-based dimension reduction

We first standardize the main batting statistics and apply PCA to them. Instead of only looking at the variance explained by each component, we fit a sequence of Huber regression models using the first $k = 1, \ldots, 6$ principal components and record several model diagnostics (RMSE, R^2, adjusted R^2, and AIC). This allows us to see how predictive performance changes as we increase the number of retained components.

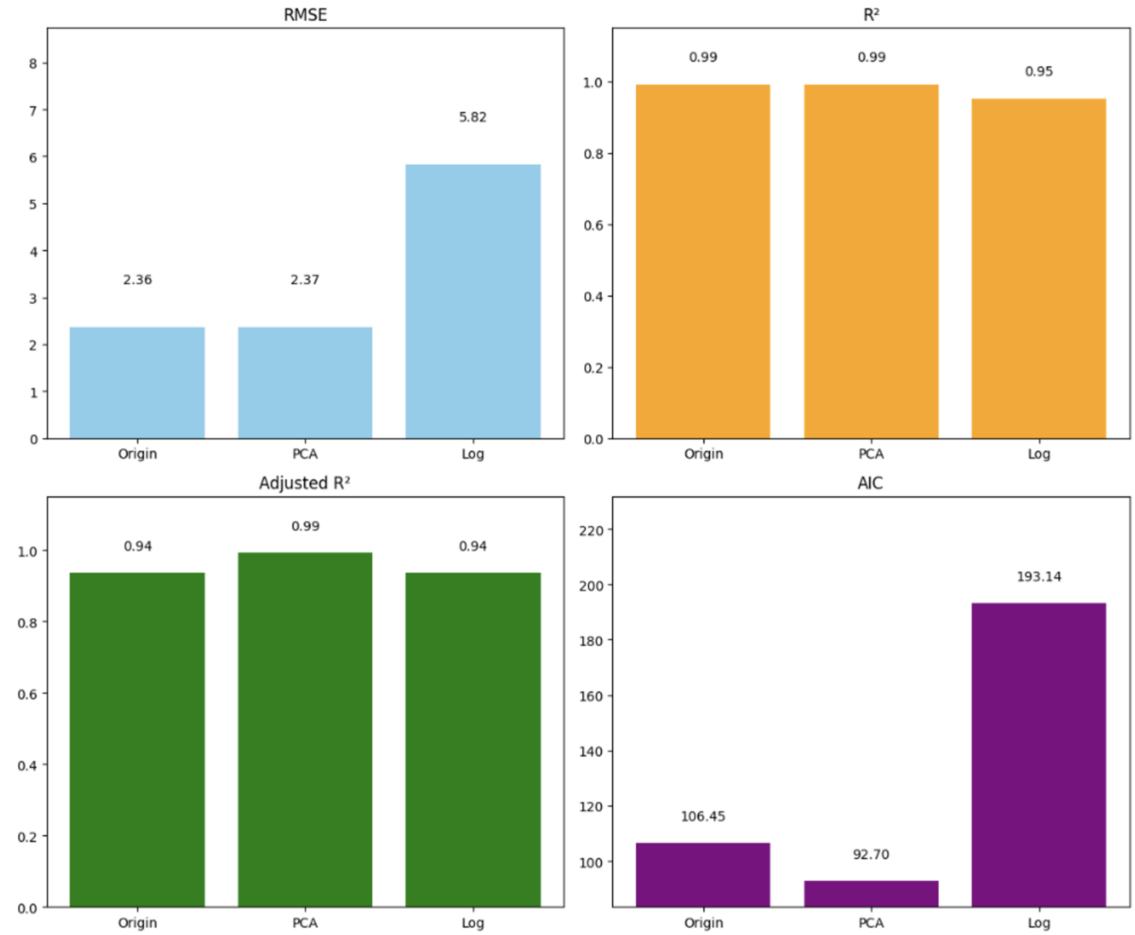


Figure 3: Model diagnostics (RMSE, R^2, adjusted R^2, and AIC) for Huber regression models using the first 1–6 principal components.

Figure 3 shows that RMSE and AIC drop sharply when moving from one to two components and continue to improve up to about five components, while R^2 and adjusted R^2 quickly level off close to 0.99. Based on this plot, we keep the first five principal components in the PCA-based Huber model.

3.2 Logistic regression for elite vs. non-elite hitters

3.2.1 Outcome and predictors

For logistic regression we define a binary outcome:

$$\text{elite}_i = \begin{cases} 1, & \text{if OPS+}_i \geq 110, \\ 0, & \text{if OPS+}_i < 110. \end{cases}$$

We restrict to hitters with at least 50 PA (about 100 players).

We use process-based predictors from `cpbl_logistic.ipynb`:

- PA
- BB_pct
- K_pct
- BABIP
- BIP_pct
- PutAway_pct

All predictors are standardized before modeling.

3.2.2 Two implementations

We use two complementary implementations:

1. **scikit-learn** `LogisticRegression` with a 70/30 train–test split to evaluate generalization performance.
2. **statsmodels** `Logit` on the full sample, first with all six predictors and then a reduced model with

$$\{\text{PA}, \text{K\_pct}, \text{BABIP}, \text{BIP\_pct}\}$$

selected using p-values.

We report training and test accuracy, confusion matrices, classification reports, and ROC/AUC.

4 Results

4.1 Robust regression results

4.1.1 Original data vs. PCA representation

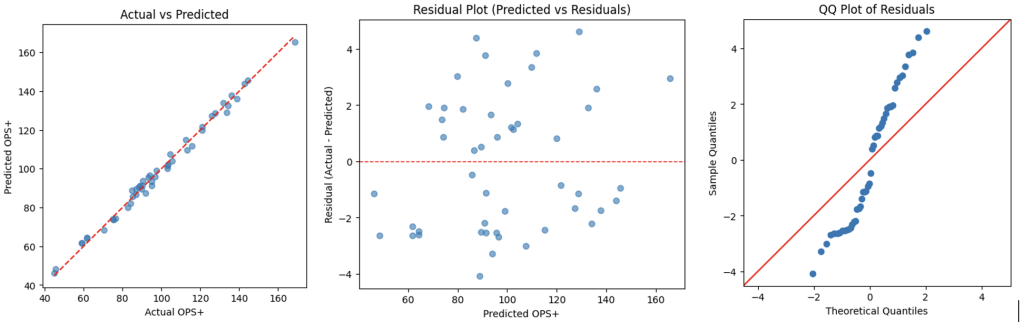


Figure 4: Robust regression using original features (example fit).

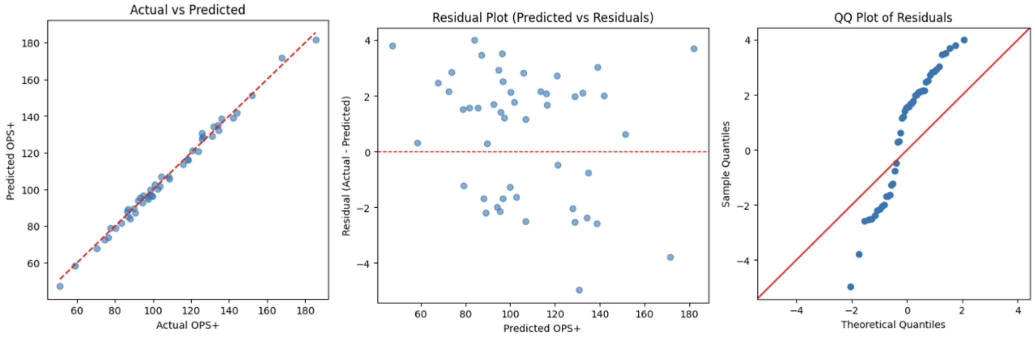


Figure 5: Robust regression using PCA (95% variance) features.

The PCA-based model achieves similar or better performance than the model on original features, while using fewer predictors and showing more stable residuals.

4.1.2 Effect of log transformation

We also experimented with a log transformation of OPS+ for robust regression. However, the log-transformed model produced worse fits and did not improve interpretability, so we kept OPS+ on its original scale.

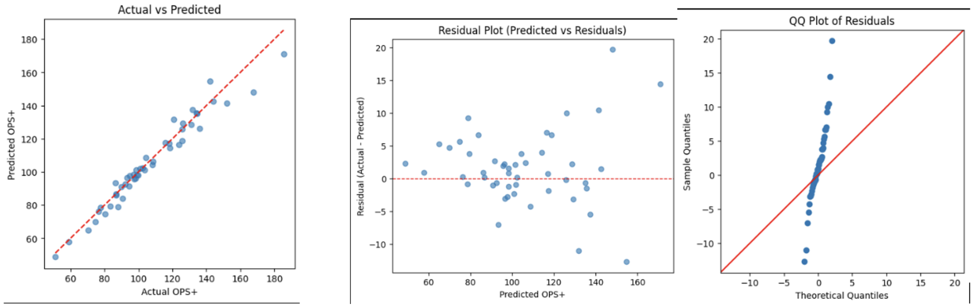


Figure 6: Effect of log-transforming OPS+ in robust regression.

4.1.3 Comparison and feature importance

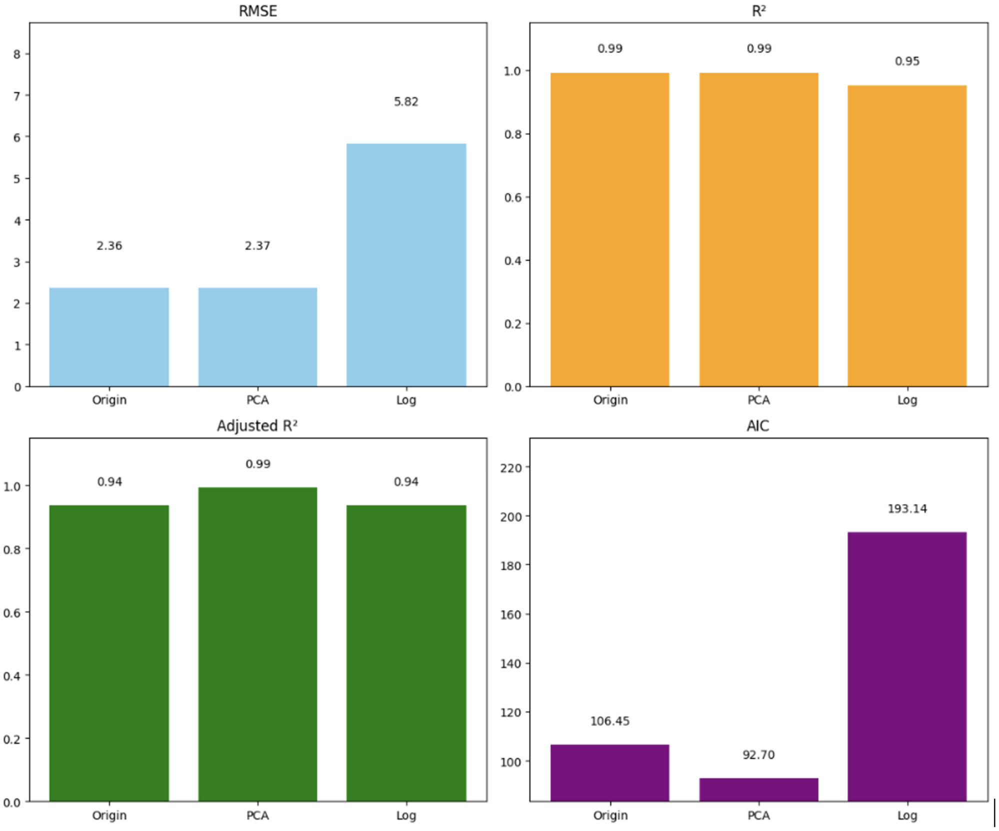
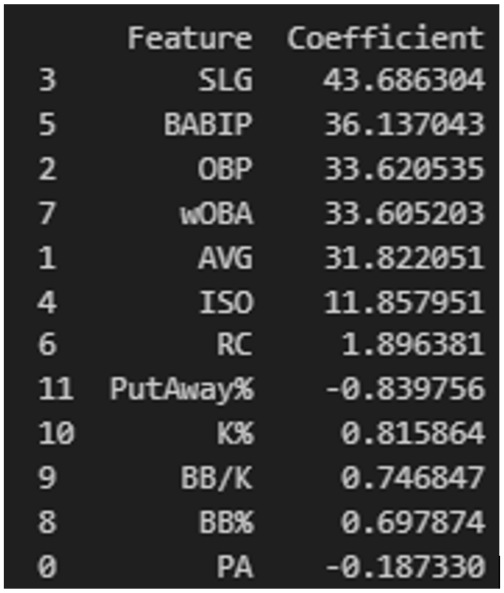


Figure 7: Comparison of robust regression models (original vs. PCA vs. log).



```
    Feature  Coefficient
3        SLG    43.686304
5      BABIP    36.137043
2        OBP    33.620535
7       wOBA    33.605203
1        AVG    31.822051
4        ISO    11.857951
6         RC     1.896381
11  PutAway%    -0.839756
10        K%     0.815864
9       BB/K     0.746847
8        BB%     0.697874
0         PA    -0.187330
```

Figure 8: Estimated feature weights from the robust regression model.

The feature-weight plot indicates that slugging percentage (SLG) is one of the most important predictors of OPS+, which is consistent with baseball intuition: extra-base hits and power strongly drive OPS+.

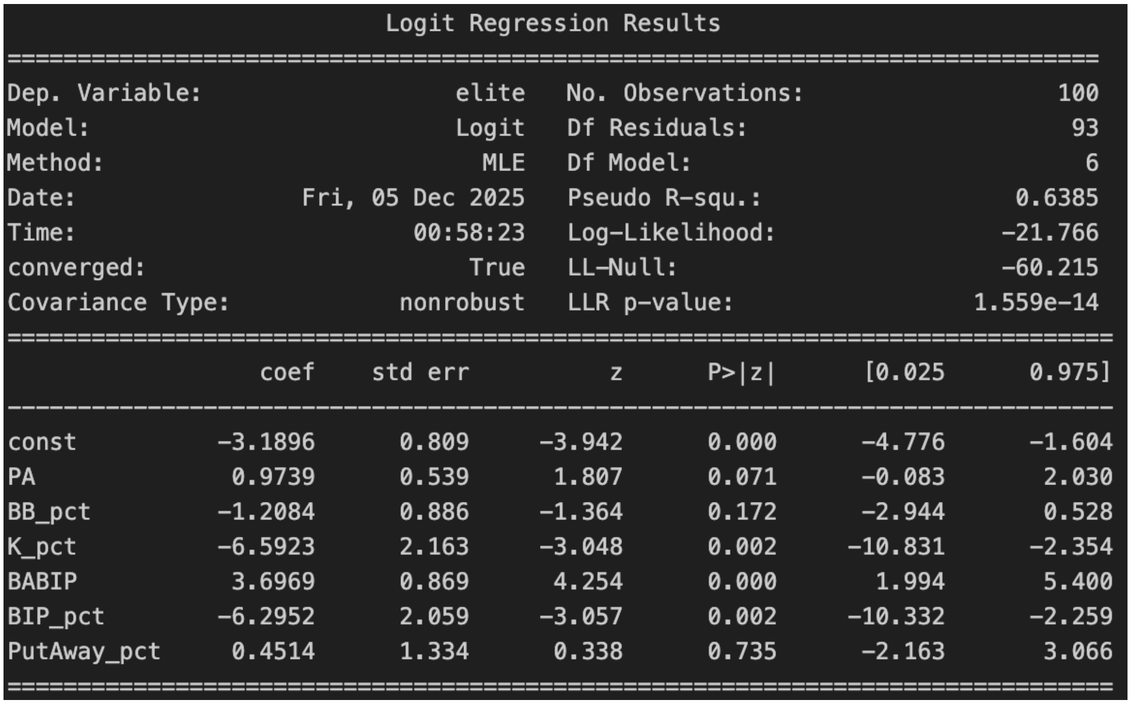
4.2 Logistic regression results

4.2.1 Model building and variable selection

We start from a full logistic regression model with all six process-based predictors:

$$\{\text{PA}, \text{BB\_pct}, \text{K\_pct}, \text{BABIP}, \text{BIP\_pct}, \text{PutAway\_pct}\}.$$

Using `statsmodels Logit` with standardized predictors and an intercept, the full model attains a high pseudo-R^2 of about 0.64 and the likelihood ratio test strongly rejects the null of no covariate effects. The coefficient table in Figure 9 shows that PA, K_pct, BABIP and BIP_pct are important, while BB_pct and PutAway_pct have much larger p-values (around 0.17 and 0.74).



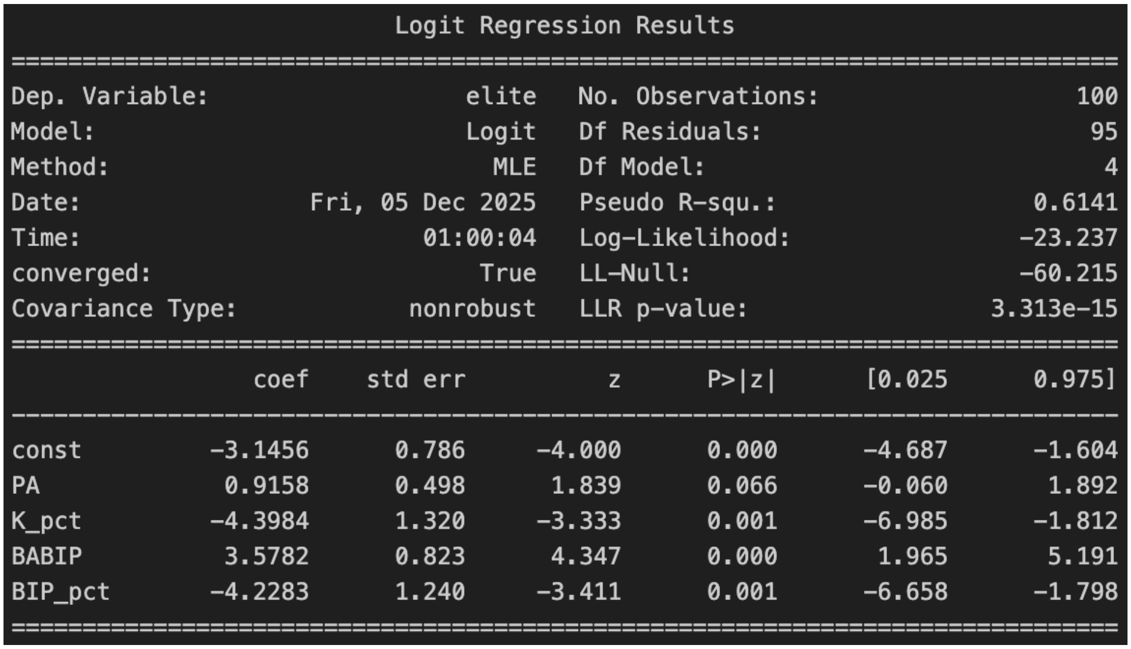
```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  elite   No. Observations:                  100
Model:                          Logit   Df Residuals:                       93
Method:                           MLE   Df Model:                            6
Date:                Fri, 05 Dec 2025   Pseudo R-squ.:                  0.6385
Time:                        00:58:23   Log-Likelihood:                -21.766
converged:                       True   LL-Null:                       -60.215
Covariance Type:            nonrobust   LLR p-value:                 1.559e-14
===============================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
const          -3.1896      0.809     -3.942      0.000      -4.776      -1.604
PA              0.9739      0.539      1.807      0.071      -0.083       2.030
BB_pct         -1.2084      0.886     -1.364      0.172      -2.944       0.528
K_pct          -6.5923      2.163     -3.048      0.002     -10.831      -2.354
BABIP           3.6969      0.869      4.254      0.000       1.994       5.400
BIP_pct        -6.2952      2.059     -3.057      0.002     -10.332      -2.259
PutAway_pct     0.4514      1.334      0.338      0.735      -2.163       3.066
===============================================================================
```

Figure 9: Summary output for the full logit model with six process-based predictors.

Based on these p-values, we fit a reduced logit model that keeps only

$$\{\mathrm{PA}, \mathrm{K\_pct}, \mathrm{BABIP}, \mathrm{BIP\_pct}\}.$$

The reduced model has a pseudo-R^2 of about 0.61, slightly lower than the full model but still high, and it achieves similar log-likelihood and better parsimony. Dropping BB_pct and PutAway_pct therefore does not meaningfully hurt the fit but makes the model simpler and easier to interpret. This four-variable specification is the one we use for both the train–test evaluation and the final interpretation; its summary output is shown in Figure 10.



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  elite   No. Observations:                  100
Model:                          Logit   Df Residuals:                       95
Method:                           MLE   Df Model:                            4
Date:                Fri, 05 Dec 2025   Pseudo R-squ.:                  0.6141
Time:                        01:00:04   Log-Likelihood:                -23.237
converged:                       True   LL-Null:                       -60.215
Covariance Type:            nonrobust   LLR p-value:                 3.313e-15
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -3.1456      0.786     -4.000      0.000      -4.687      -1.604
PA             0.9158      0.498      1.839      0.066      -0.060       1.892
K_pct         -4.3984      1.320     -3.333      0.001      -6.985      -1.812
BABIP          3.5782      0.823      4.347      0.000       1.965       5.191
BIP_pct       -4.2283      1.240     -3.411      0.001      -6.658      -1.798
==============================================================================
```

Figure 10: Summary output for the reduced logit model with four predictors (PA, K_pct, BABIP, BIP_pct).

4.2.2 Train–test performance (scikit-learn, 4 features)

Using the reduced feature set $\{\mathrm{PA}, \mathrm{K\_pct}, \mathrm{BABIP}, \mathrm{BIP\_pct}\}$ and a 70/30 train–test split, we fit a `LogisticRegression` model in `scikit-learn` with `max_iter = 1000` and `class_weight = "balanced"`. This model yields:

- Training accuracy: about 0.83.
- Test accuracy: about 0.87.
- Test AUC: around 0.93.

On the 30% test set, the confusion matrix (rows = true class, columns = predicted class) from the notebook is:

Table 2: Test-set confusion matrix (scikit-learn logistic regression, 4 features).

	Predicted 0	Predicted 1
True 0	19	2
True 1	2	7

This corresponds to correctly classifying most non-elite hitters and the majority of elite hitters. The ROC curve on the test set is shown in Figure 11.

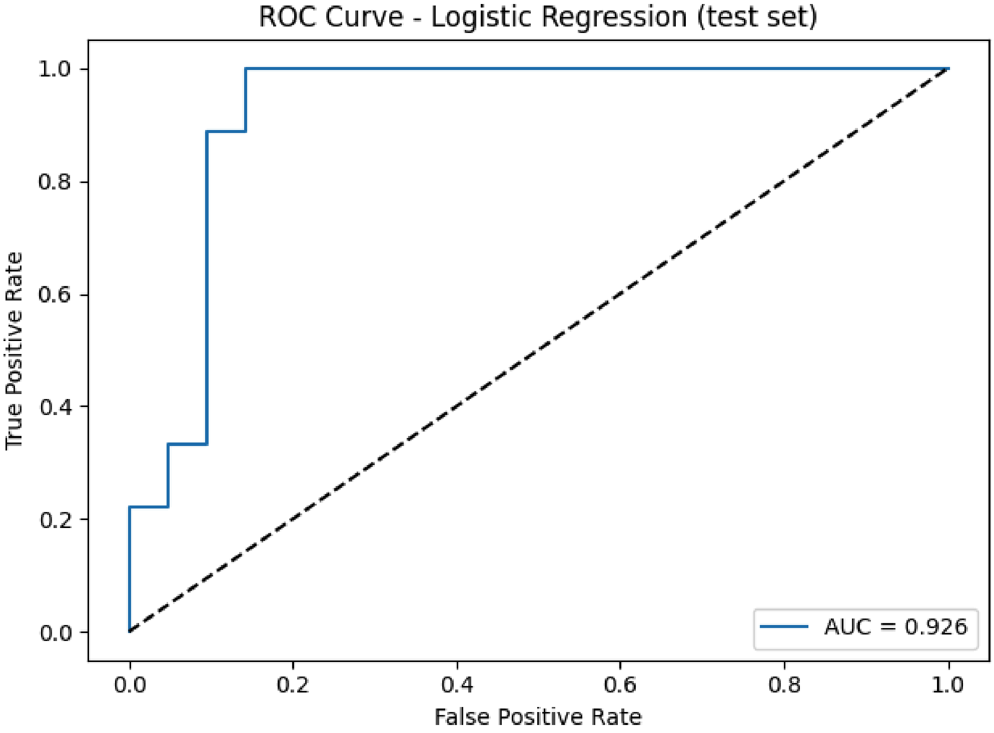


Figure 11: ROC curve for the reduced logistic regression model on the test set.

4.2.3 Full-sample inference (statsmodels reduced logit)

For more detailed inference, we return to `statsmodels` and fit the reduced logit model with the same four predictors on the full sample of about 100 hitters. Using a probability threshold of 0.5 to classify elite vs. non-elite, the confusion matrix is:

Table 3: Full-sample confusion matrix (statsmodels reduced logit).

	Predicted 0	Predicted 1
True 0	66	5
True 1	6	23

This corresponds to an in-sample accuracy close to 0.89. The ROC curve in Figure 12 shows that the in-sample AUC is also above 0.9, indicating strong separation between elite and non-elite hitters on the full data.

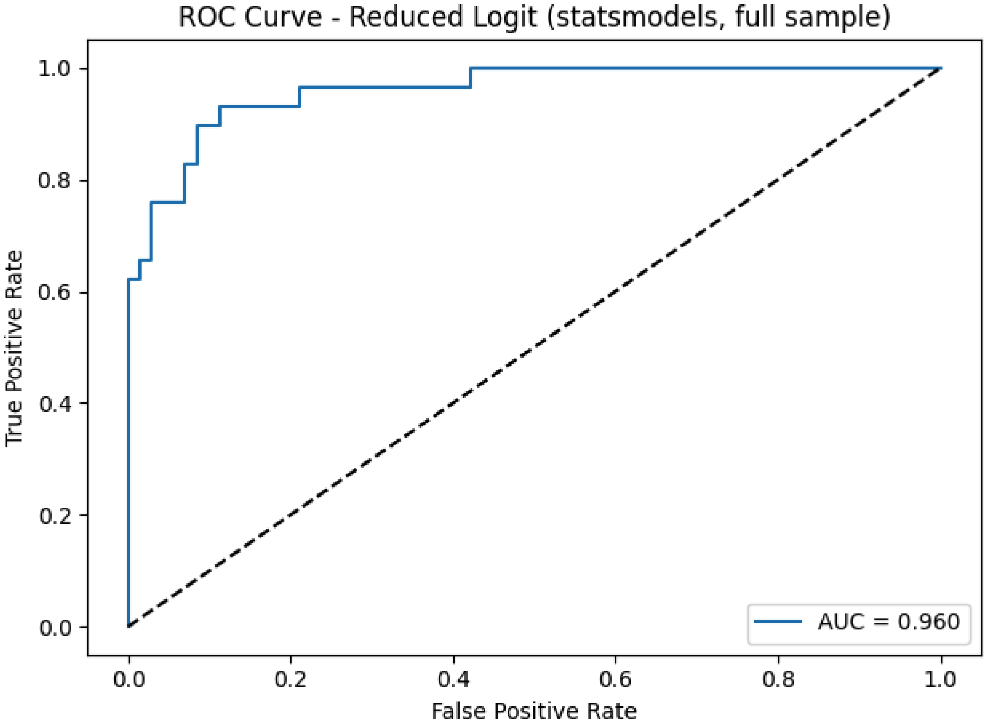


Figure 12: ROC curve for the reduced statsmodels logit on the full sample.

We also convert the estimated coefficients into odds ratios via

$$\mathrm{OR}_j = \exp(\hat{\beta}_j),$$

so that they can be interpreted on the standardized scale:

- A one-standard-deviation increase in PA multiplies the odds of being elite by a factor larger than 1.
- A one-standard-deviation increase in K_pct substantially reduces the odds.
- A one-standard-deviation increase in BABIP multiplies the odds by a large factor.
- Higher BIP_pct is associated with lower odds of being elite.

5 Discussion

The robust regression and logistic regression analyses tell a consistent story about what characterizes strong CPBL hitters.

From the robust regression analysis, slugging percentage (SLG) stands out as a key driver of OPS+, which matches basic baseball intuition: power and extra-base hits matter a lot. The PCA-based model shows that we can compress much of the information in the batting statistics into a small number of components without losing too much predictive power, and that this can also stabilize the fit in the presence of outliers.

The logistic regression analysis deliberately uses process-based statistics that do not trivially encode OPS+ itself. It shows that:

- Low strikeout rates (low K_pct) and high BABIP are both strong predictors of being an elite hitter.
- Plate appearances (PA) are positively associated with elite status, which likely reflects both talent and opportunity.
- A high BIP_pct is associated with lower odds of being elite, possibly because simply putting more balls in play without strong plate discipline or power does not create enough high-value outcomes such as walks and extra-base hits.

On the test set, the logistic regression model still achieves high AUC and good accuracy, which suggests that these four process-based features generalize reasonably well beyond the training sample.

6 Conclusion and Future Work

To sum up, this STT 810 project shows that:

1. Robust regression with PCA provides a stable and interpretable model for continuous OPS+, and highlights the central role of slugging percentage and related power metrics.

2. Logistic regression based on process-based statistics (PA, K_pct, BABIP, BIP_pct) can successfully distinguish elite from non-elite CPBL hitters, with good out-of-sample performance and high AUC.

Several extensions are natural:

- Add more seasons and data sources to increase sample size and check whether the patterns we see here are stable over time.
- Use cross-validation for more systematic model selection and tuning.
- Compare with regularized and non-linear models (for example, Lasso, random forests, gradient boosting) as benchmarks.
- Extend the analysis to position-specific models or to pitcher performance metrics.

All code, intermediate data, and notebooks for this project are available at:

https://github.com/neil7227/STT810-project/tree/main

References

References

[1] Robas (*The Baseball Revolution*) CPBL batter statistics, 2024–2025 seasons.

[2] Scikit-learn developers. *Scikit-learn User Guide: Linear and Logistic Regression.*

[3] Statsmodels developers. *Statsmodels User Guide: Discrete Choice Models.*