Pythagorean Expectation as a Predictor in Baseball

Erdos Institute Fall 2023
Steven Creech, Mohsen Mohaghegh, Joe Schulze,
Weiqi Wang, Billy Warner
Github Repository Link

Introduction

- Pythagorean Expectation is a rather simple formula that gives a fairly close approximation for the win percentage of a baseball team the formula states:

$$\text{Win Percentage} \approx \frac{RS^2}{RS^2 + RA^2}$$

- Our goal was to build a predictive model using the idea of the Pythagorean Expectation, we built two models and compared them with an Elo model by 538
- The metric we choose to use to compare models is the so called Brier score:

$$\text{Brier Score} = \frac{1}{n}\sum_{i=1}^{n}(p_i - o_i)^2$$

Regression Model

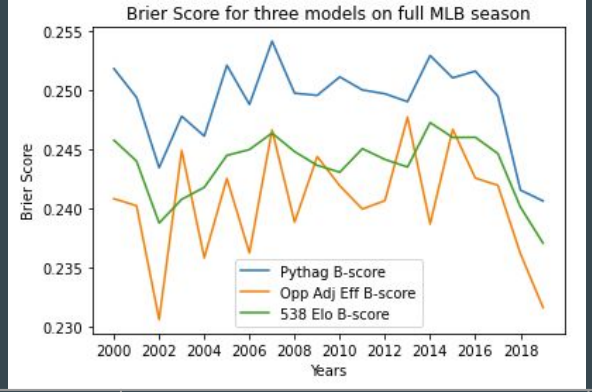
- This model computes the probability of a team winning by a logistic regression on one statistical feature representing team strength.
- The feature that we analyze is a team's opponent adjusted efficiency for runs scored RS_{adj} and runs allowed RA_{adj}.

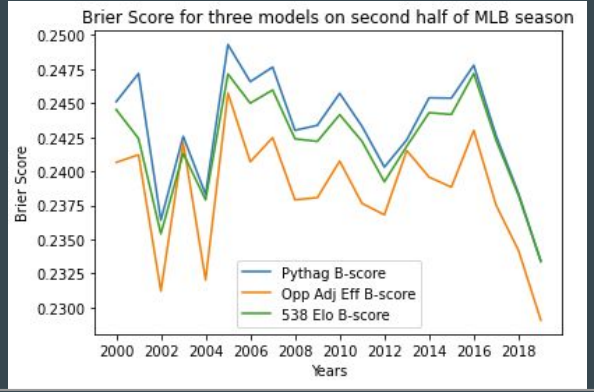
$$RS_{adj} = (1/n)\sum_n (RS_n/RA_{n,avg} - 1) \qquad RA_{adj} = (1/n) \sum_n (1 - RS_{n,avg}/RA_n)$$

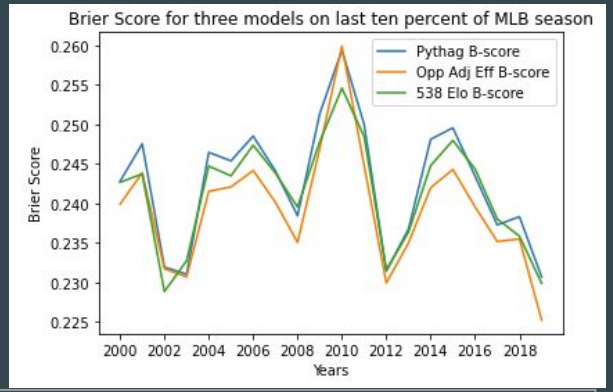
RS_n = Runs Scored in game n $RA_{n,avg}$ = Average Runs Allowed by Opp up to game n

RA_n = Runs Allowed in game n $RS_{n,avg}$ = Average Runs Scored by Opp up to game n

- We compare this model to the Pythagorean Expectation and 538's Elo model.







Opp Adj	0.2404 ± 0.001 B-Score	0.2385 ± 0.0009	0.2394 ± 0.0016
Pythag	0.2490 ± 0.0008	0.2432 ± 0.0009	0.2426 ± 0.0017
538 Elo	0.2436 ± 0.0006	0.2421 ± 0.0008	0.2413 ± 0.0015
Opp Adj 538 Elo	0.595 Correlation Coef	0.914	0.943
Opp Adj Pythag	0.704	0.927	0.972
Pythag 538 Elo	0.920	0.973	0.978

Prediction via Simulation

- Consider a match between team (x, n_x) and team (y, n_y)
- Principles

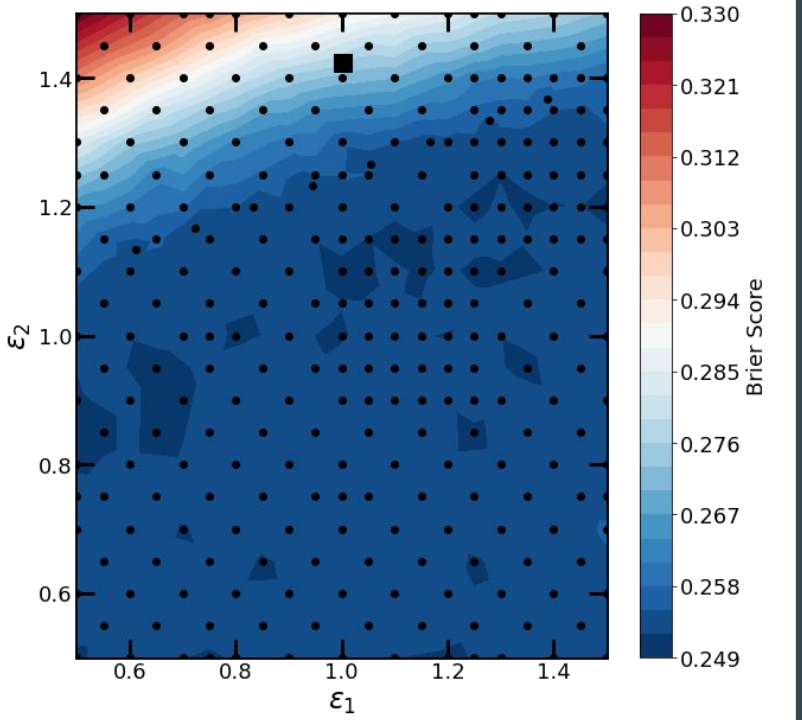
$$\begin{cases} R_s^x \sim Weibull(Scale_x, Shape) \\ \\ |PE_{n+1} - PE_n| \leq \delta_n \end{cases}$$

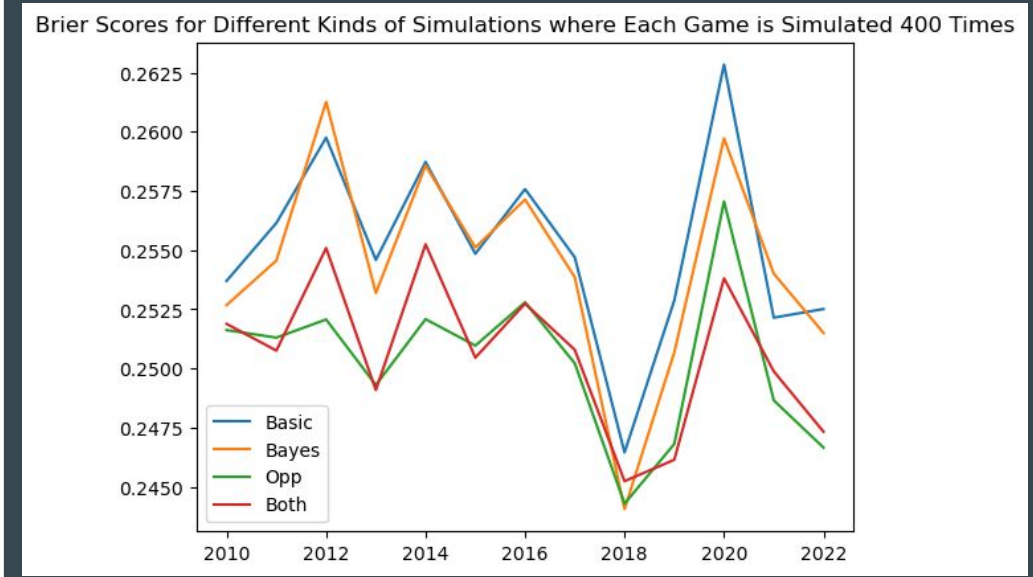
  - Empirically-determined parameters

- Algorithm:
  - For each possible game, draw R_s^x and R_s^y
  - "accept" only if the implied change in PE is less than δ_n
  - Simulate N acceptable games, and use to predict

Optimization of ε_1 and ε_2

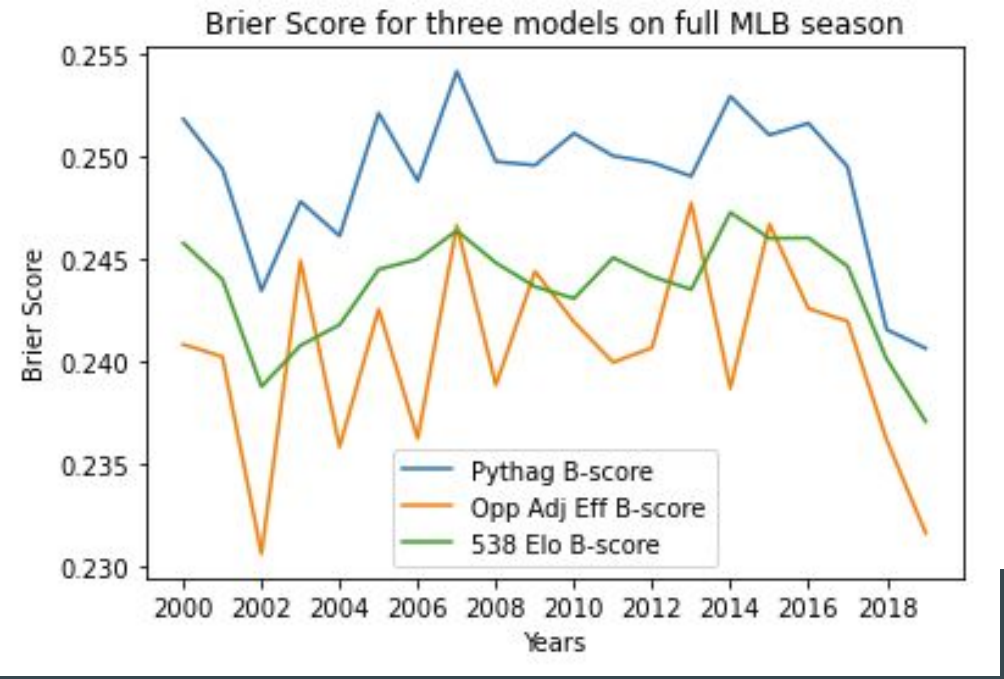
$$\ln \delta_n = \ln \varepsilon_1 - n \ln \varepsilon_2$$





Conclusion

Regression Model



Simulation Model