ST 491 Midterm Writeup

Tyson King

2023-03-27

Exploratory Data Analysis

The dataset that I am going to be making inferences on contains information on NCAA softball games during the 2022 and 2023 seasons. I am planning on using the 2022 season as the training data set and make inferences on the 2023 games. I am going to create a multiple linear regression model that uses a simple ranking system (RPI) and make conclusions on the run differential between the two teams in a given game. The RPI ranking system generates a ranking coefficient between 0 and 1 with the formula:

$$RPI = 0.5 * (win\%) + 0.25 * (opponents'\ win\%) + 0.25 * (opponents'\ opponents'\ win\%)$$

Here is a link to the dataset on GitHub:

https://raw.githubusercontent.com/tmking2002/st491_midterm_project/main/scoreboard_dataset.RDS

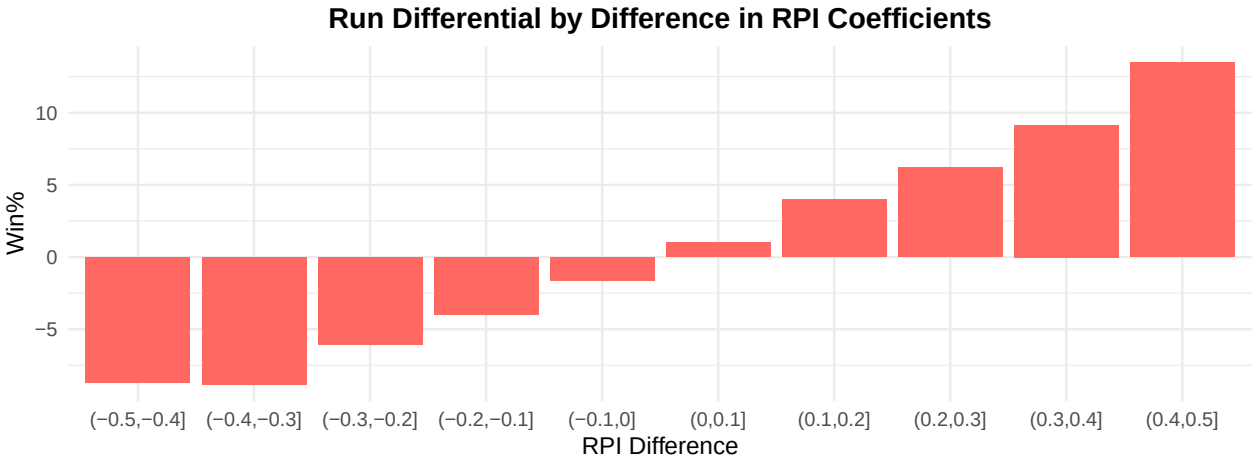
This is the model I will be making:

$$run\ differential = \beta_0 + \beta_1 team1\_rpi + \beta_2 team2\_rpi$$

Here is a small snippet of the dataset:

date	team1	team1_rpi	team2	team2_rpi	team1_runs	team2_runs	score_diff
02/09/2023	Long Beach St.	0.5427829	Arizona	0.6203258	1	9	-8
02/09/2023	Saint Joseph's	0.4627947	Boston U.	0.6315283	0	8	-8
02/09/2023	North Carolina	0.4836730	BYU	0.5892880	2	0	2
02/09/2023	North Carolina	0.4836730	California Baptist	0.5464802	3	2	1
02/09/2023	Merrimack	0.4340464	Charleston So.	0.4405469	0	3	-3
02/09/2023	South Carolina	0.6455717	Charlotte	0.5933963	1	9	-8

Generally speaking, the team with the higher RPI coefficient is more likely to win any given game. The question that I'll be trying to answer is *how much* this difference impacts the expected run differential between the teams.



Generate Synthetic Data

I used the sample mean and sample standard deviations to create Gaussian distributions for team1_rpi and team2_rpi with n = 10000. Then, for each row of rpi values, I created predicted run differential using a very crude technique (10 * team1_rpi - 10 * team2_rpi).

Determining a Statistical Method for Estimating Parameters

$$run\ differential = \beta_0 + \beta_1 team1\_rpi + \beta_2 team2\_rpi$$

Now, I'll use least squares estimation to find the optimal values for β_0, β_1, and β_2 from the synthetic data set.

True β_0: 0

Estimated β_0: 0

True β_1: 10

Estimated β_1: 10

True β_2: -10

Estimated β_2: -10

Evaluate Model on Real Dataset

Estimated β_0: 0

Estimated β_1: 27

Estimated β_2: -27

Final Model from LSE: Run differential = 0 + 27 * team1_rpi - 27 * team2_rpi

Mean Squared Error: 19.563