Leveraging AI To Automate MLB Roster Construction; Using High Fidelity Synthetic Data To Optimize Team Performance

Parker Gibbons

IPHS 391 Frontiers in Generative AI (Fall 2024) Prof Chun, Kenyon College

Abstract

This project utilizes artificial intelligence chatbots to generate high-fidelity synthetic data for Major League Baseball hitters and their statistics over a single 162 game season. The AI evaluates each player based on a five-tool scale, comparing their abilities to one another. Finally, optimization techniques are applied to select an optimal set of thirteen hitters (half of a full MLB roster of twenty-six players), maximizing overall performance while adhering to a specified salary cap. The operating hypothesis is that artificial intelligence can effectively generate high-fidelity synthetic data for baseball hitters and evaluate player performance based on a five-tool scale, given a large influx of data. By applying optimization techniques, it is possible to construct a competitive baseball team that maximizes overall performance while staying within a given salary cap, mirroring real-world decision-making in sports management.

Introduction

In Major League Baseball, players are evaluated using a variety of methods. While traditional statistics such as batting average, home runs, runs batted in, and stolen bases remain important, newer sabermetrics are playing an increasingly significant role in the evaluation process. These advanced metrics have transformed scouting and player assessment. With the growing prevalence of artificial intelligence, this process could evolve even further. AI introduces technological capabilities that reduce human bias and automate processes widely done today, allowing for more objective and efficient evaluations. By training models on large datasets and information, AI can comprehensively assess player abilities, providing Major League Baseball organizations with valuable insights for making informed decisions. Making this process more automated and easier to perform could enable individuals to increase the amount of analysis they can do single handedly by reducing the time needed to complete tasks such as evaluation. This could make it easier for organizations to construct their teams while placing emphasis on certain attributes or statistics.

This project focuses solely on hitters (about one half of the players in Major League Baseball). Historical data of Major League Baseball hitting statistics and contract information are fed into the AI chatbots ChatGPT 4o and Claude AI in order to create a dataset of hitters to be used for the subsequent evaluations and roster optimization. ChatGPT-4, developed by OpenAI, is based on the transformer architecture and trained on extensive datasets consisting of diverse text from books, websites, and other publicly available sources, ensuring a broad understanding of language and context. Similarly, Claude AI, created by Anthropic, is engineered with a focus on safety and reliability, leveraging a reinforcement learning process that aligns its behavior with user instructions and ethical considerations. Both models are designed to perform tasks such as drafting content, answering questions, coding, summarizing information, and providing language support, making them invaluable tools for productivity and automation.

Methodology

In order to begin the process, I first had to research in order to find data that best encapsulated the type of dataset that I wanted to create using the AI chatbots. The first source of data that I decided to use was Baseball Savant which is a website that provides player matchups along with statcast metrics, and advanced statistics in a user friendly way. I also used sources such as Baseball Reference and Fangraphs which both serve as statistical databases for any player in the history of professional baseball, along with a wide range of sabermetrics used to further analyze player performance. I also used Sportrac to obtain information of team payrolls and player contracts in Major League Baseball. These sources were all used to inform and limit the AI on what data to value and utilize during its data generation and evaluations.

The next step was to utilize prompt engineering techniques with the AI Chatbots to make the dataset more realistic. This process involved feeding the AI Chatbots data for players at each of the positions (excluding pitchers) in order to serve a guide when generating the synthetic data. This data including statistics and metrics on hitting, fielding, throwing, and speed. This would help make the players at each position have data more accustomed to those who play the same position in real-life. This process also involved splitting up the hitters by position according to the actual positional distribution within Major League Baseball, making the statistics mathematically consistent (e.g. Batting average = Hits/At-bats), and producing realistic yearly salaries for each player given their performance statistics. Generating salaries also accounted for the idea that about 38% of Major League hitters earn a yearly salary at or extremely close to the league minimum salary of $740,000. Some statistics chosen to highlight player offensive output were On-base Plus Slugging Plus (OPS+), Weighted Runs Created Plus (wRC+), and Isolated Power (ISO).

Once the synthetic dataset was generated, I tasked the AI chatbots with evaluating each player according to a five-tool scale (Hitting, Hitting for power, Fielding, Throwing, Speed) and supplying an inclusive overall rating. These ratings spanned from 20-80 to reflect the baseball industry standard for grading players and their tools. Using these new ratings, I created a program that has the ability to compare any two players in the dataset to one another based on their five tools and overall rating. I then used these ratings and comparisons to produce brief scouting reports for each player and their tools, which I visualized using Gradio, a Python package used to build webapps for machine learning models, APIs, or Python functions.

The last step of the process was creating a function to optimize a set of thirteen hitters in the dataset to have the best overall performance (calculated using the sum of previously calculated overall ratings for each player). This selected set of players had to adhere to the constraints of a salary cap, a minimum of one player at each position, and a maximum of two players at each position. To do this, I used the PuLP library, a tool for linear programming (LP) and integer programming problems. PuLP allows users to define, solve, and analyze optimization problems where the objective is to minimize or maximize a linear function, subject to linear equality and inequality constraints. This allows the optimization process to work regardless of the salary cap amount.

Results

Synthetic Data Generation:

The AI chatbots were quickly able to produce an initial dataset but it took a great deal of prompt engineering to increase the fidelity of the data. The chatbots had to be reminded multiple times to make the data mathematically consistent and needed to be fed formulas for specific statistics such as WRC+. The chatbots specifically struggled with calculating a WAR estimate for each player because of the lack of universal public formula used. In addition, the AI struggled with creating exceptions to the various positional attributes that I fed to them. It took a while to make sure that not all of the first basemen, shortstops, etc, had the same attributes. The chatbots also struggled with reflecting the mostly normal distribution of certain statistics, including average, WRC+ and WAR, which led to the tails of the distribution being much heavier than they actually are.

Evaluating Players:

The AI chatbots did a fairly good job at evaluating each player based on the given metrics. However, the calculated ratings did not do very well at comparing to the 20-80 ratings for players widely used in the baseball industry. This could have resulted from the AI struggle in capturing the distribution of stats among the dataset. Nevertheless, the AI made it very simple and easy to compare players to one another and determine who is superior in each of the five tools and overall. The overall rating made it easy to get the multitude of metrics for each player down to a singular number to evaluate them by.

Optimizing Roster Construction:

Using the PuLP library, the player ratings previously calculated were able to be used to optimize a set of thirteen hitters while adhering to positional constraints and a maximum salary cap. With a lower salary cap (e.g. 20 million dollars), the model largely constructed a roster full of younger players being paid at or just above the league minimum. With a large salary cap (e.g. 300 million), the model selected more expensive players to obtain a higher team performance rating because of the increased spending flexibility. Using the data given, the model prioritized spending on a center fielder, catcher, and shortstop. However, even with a large salary cap, the model often sought out the less expensive players to minimize the amount of salary spent on players.

20 million cap	Metric	200 million cap
10,317,079	Money Spent	122,957,079
6,271	Performance	6,321
1,683,959	C$	42,943,959
740,000	1B$	740,000
1,480,000	2B$	740,000
1,733,061	3B$	1,683,959
740,000	SS$	25,000,000
1,480,000	LF$	1,480,000
740,000	CF$	48,600,000
1,720,059	RF$	1,720,059

Conclusion

While not nearly being perfect, the AI chatbots provided an automated method for generating a mostly realistic dataset of hitters in Major League Baseball. However, the AI took much prompt engineering and data in order to make the data comparable to actual MLB data with regards to statistical distribution and positional attributes. The AI was also able to utilize statistics and metrics to calculate ratings for the tools and overall ability of each player, thus providing quantitative evaluations for every player. These ratings made it easy to optimize a roster of hitters of high performance while minimizing the amount of money spent on players. Rookie players making the minimum salary were favored because of their lower salaries as compared to higher paid veteran players.

Limitations and Future Work

The work in this project is ultimately limited to the data fed to the AI chatbots. In the future, I would extend this project by generating even more data for each player, including experience length in the MLB, physical attributes (height, weight, handedness), awards received, and draft spot. In order to do this, I would feed the chatbots even more data in order to make the data generated as realistic as possible. I would also include more than a single season of statistics to base player evaluations on and also place different weights on each previous season in the overall evaluations. I would also expand on each of the five tools to include more features for each of them in order to create more comprehensive ratings. The AI chatbots could also be used to generate pitcher data too in order to evaluate a full league of MLB players. This idea could be extended to evaluate players while placing weights on certain tools and metrics to prioritize during the evaluation process.

Within the optimization process, alternative methods of optimization could be used to see if the roster of hitters selected differs by method. In addition, specific weights and values could be placed on certain positions or statistics in order to guide the optimization with prioritized metrics. In addition, the salary cap could be constrained by position to place spending emphasis on certain positions over others.

While automating the process, this project obviously overlooks the human factor of the scouting process, which is still valued by teams. Finding a way to utilize both automated evaluations with AI and human scouting could provide a solution that captures the best aspects of both methods.

Annotated Code and References