Machine Learning Algorithm for Predicting Major League Baseball Team Wins

A Senior Project Report

presented to

the faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

Of the Requirements for the Degree

Bachelor of Science in Computer Engineering

By

Bryce Farrell

June 2019
Introduction

I was inspired to work on this project by the book “Moneyball” by Michael Lewis. The book discusses the advancements made in sports analytics, and it gave me the idea to take Major League Baseball analytics to the next level. Machine learning and artificial intelligence is the future of many industries, but analytics and statistics is a clear starting point. I decided to combine the power of neural networks with traditional baseball statistics to predict the win totals of teams. Ideally, this information would be used to inform teams on where they need to improve to secure more wins. This project lies at the intersection of sports, statistics, and computer science, which all appeal to my interests. This is also a project that I can continue working on past graduation, potentially transitioning it into a full-sized application and maybe even a product.

Project Definition

The application utilizes a neural network trained on 60 years of past baseball statistics to predict whether or not the team in question achieved a certain win total. This kind of neural network is called a binary classification model because it chooses between only two outputs: reaching win total or missing win total. The desired wins, team, and year is inputted by the user of the program. If the algorithm predicts the team to not achieve the win total, it then calculates the changes necessary to each statistic that would cause the team to win the desired amount of games. The algorithm can be used to analyze statistics of every Major League Baseball team from the years 1960 to 2018. Additionally, I included an accuracy script that tests the accuracy of the algorithm for each potential win total, outputting a graph of error percentage per win total.
Design and Results

The binary classification neural network was built using a python framework called pytorch, which does not change the python experience as much as other machine learning frameworks. To train the model, the framework requires that the programmer specifies what is called a loss function. For this, I chose a binary cross-entropy loss function that works well with binary classification networks. After the core algorithm was established, I began to gather data from a website called Baseball-Reference. There were no easy ways to collect a large historical datasheet so I was forced to collect data year-by-year. The result of the accuracy script is shown below in Figure 1.

![Figure 1: Error Rate vs. Win Totals](image)

The error rate was highest around the 70 to 80 win section due to the fact that this is the win totals where many teams end up, making it more difficult to discern between statistics. Of course, the algorithm predicted with 100% certainty the teams who won or lost a lot of games.
After I was confident the algorithm worked using my accuracy script, I moved on to the data manipulation portion. The goal was to find out how a team could reach this win total. The process of finding the necessary statistics involves a series of random changes to the input statistics and re-running the updated inputs through the neural network until the win total was met. The result of running the program with its most verbose output is shown below in Figure 2.

![Program’s Output](image)

Figure 2: Program’s Output

The bottom section of this output details the actual statistics of the team, the necessary statistics to reach the win total, and the changes needed to get there. Because there exists so many different ways the statistics could be changed to reach the desired win total, I have included the standard deviation of all the iterations of changes I made for each statistic.

Conclusion and Future Work

Machine learning and neural networks are the future of data science, artificial intelligence, and computer science in general. The technique can be used to take daunting amounts of data and seemingly difficult problems and solve them simply with little code-writing overhead. This project serves as a proof-of-concept for a neural network that can tackle the topic
of advanced sports prediction and analysis. I am excited to transition my skills learned when working on this project to more complicated applications of machine learning into my future as a software engineer.