Moneyball: Implementation of Sabermetrics and Machine Learning

|  |  |  |
| --- | --- | --- |
| **Ryan Gleeson** Computer Science Department rgleeson@oberlin.edu | **Luis Solórzano** Computer Science Department lsolorza@oberlin.edu | **Isabel Ehrhardt** Computer Science Department iehrhard@oberlin.edu |



1 INTRODUCTION

The summer blockbuster Moneyball is regarded as one of the best sport movies of all time. It featured a great backstory, excellent character development, and overcoming of adversity. The main reason why it stood out to sports fans is that it illuminated an important and relatively new tactic for assembling the best team possible. In this final report, we will first motivate our undertaking of this project, primarily influenced by our affinity for sports and computer science. Furthermore, the interplay of statistics, computer science, and baseball enables us to present a brief introduction to sabermetrics. Then, an overview on the types of data which we plan to use and some of their pitfalls will be presented. After, we will give a report of the work taken to create a solution to certain sabermetric problems, including the unique adversities we faced in this project. We will also focus on the future implications of our work and how this type of computer programming may change scouting and baseball in general. Finally, we will cover ideas that we would want to pursue in order to further develop this program. Baseball has a huge intersection with statistics, which is obvious even in basic stat lines such as batting average and earned run average. The motivation behind moneyball comes from wanting to maximize a baseball team’s competitiveness. In major league baseball, there is no salary cap per club. Consequently, some teams are able to offer players more money than others, which often creates an unfair advantage. But because baseball is a slow-paced sport, any team is able to use statistics to their advantage. For example, teams will align their infield and outfield based off of the hitter’s patterns when putting a ball into play. Sometimes batters are more prone to hit towards their batting stance or cut the ball away, so the defense will align themselves so that they’re in a better position to get a batter out. In the movie Moneyball, Billy Beane (A’s general manager) brought in players that the traditional valuation system had discarded. He was able to see their potential and get them on his roster for less than they may have been worth. The new valuation methods gave him insights that were not available to those using the old methods. These methods started a revolution on how to measure the utility of a baseball player that have continued to gain popularity in baseball and other sports. The techniques that we learn and implement in this project will be some of the stepping stones that propel sports into a new, data-driven age. One of the most important of the analytical stepping stones is sabermetrics, which we will be using as a reference in this project.

2 Problem

2.1 Brief Introduction to Sabermetrics

Sabermetrics is a science, which means that it follows the scientific method. Conclusions must be based on evidence and logic, and any conclusions can be reevaluated or overturned if new, contradictory, evidence turns up. Sabermetric researchers often use statistical analysis to question traditional measures of baseball evaluation such as batting average and pitcher wins. Rather than weighing a player’s productivity based off of batting average or earned run average (ERA), statisticians have created advanced metrics such as wOBA(weighted on-base average), FIP (fielding-independent pitching) and WAR (wins above replacement)[1]. Baseball statistics are designed to answer questions such as “who was the best base runner in the American League?” Statistics allow us to gather data points from individual events and summarize them in ways that are easy to understand. When the use of ‘moneyball’ began it was seen as an attack against scouts and their merit. Now virtually every team uses advanced statistical analysis to evaluate players. The question for teams now is, “how best can we apply the principles of Moneyball?”.

2.2 Description of Data

All teams have access to the same data, so it is critical for teams to discover and design methods to use the data to their advantage. When building a team we will look at different statistics for pitchers and position players (non-pitchers) [2]. For position players, we will focus on their production at the plate.

|  |
| --- |
|  |
| ***Figure 1.*** |

**Figure 1** shows the statistics we analyzed for position players. They are standard statistics that would show up on the back of a baseball card. Some of the less known ones are on base percentage (OBP), which looks at the percentage of a player being on base whether through a walk, hit by pitch, or hit. Slugging percentage (SLG) takes the Total Bases/At Bats or (1B + 2\*2B + 3\*3B + 4\*HR)/AB. Rather than looking at each individual type of hit whether it be a base hit, double, triple or homerun, SLG makes it easier to look at all these statistics in a given context.

In **Figure 1**, one of the first flaws that stands out is that some players have incomplete statistics. Incomplete instances were removed from our program. Next, we had to make sure each player’s position (POS) was recorded. POS featured the enumeration of position with other symbols that gave more detail such as a ‘D’ for designated hitter or ‘\*’ indicating that the player played at least ⅔ of their team’s games.

|  |
| --- |
|  |
| ***Figure 2.*** |

For each player’s position in an instance object, we choose the leading number which was the player’s primary position. Another bug in the data was that some players have the same name, or players played on different teams in a given year. Luckily, baseball reference sources enabled easy, specific identification of players despite these complications. In order to handle players that played on multiple teams, we only took the statistics for a given identification once. While a pitcher’s offensive production does matter if the team is in the National League, we will ignore it, like many other teams do, since a pitcher’s performance on the mound is far more important than his production at the plate. With a pitcher we can start by dividing aspects of run prevention into two categories: those that the pitcher controls almost entirely and those in which his defense plays a major role. Pitchers should be held accountable for strikeouts, walks and home runs. On the other hand, hits are partially dependent on factors outside the pitcher’s control. The pitching instances can be seen above in **Figure 2**. Like the position players, there are repeats instances of players, which we handled in the same way. The stats at the end of the instance don’t represent the pitchers offensive production. Instead, they are the count of hits allowed (H), home runs allowed (HR), walks allowed (BB), and strikeouts (SO). For both pitchers and position players we used wins above replacement (WAR) as a label for our prediction value**.** WAR is an attempt to summarize a player’s total contributions to their team in one statistic**.** WAR is not meant to be a perfectly precise indicator of a player’s contribution, but rather an estimate of their value to date. Given the imperfections of some of the available data and the assumptions made to calculate other components, WAR works best as an a single-value summary to use as a label. To calculate WAR for pitchers, statisticians use runs allowed as their base, but also works in a control for defense after the fact. Batting average on balls in play(BABIP), walk rate, home run for fly balls (HR/FB%), expected fielding independent pitching (xFIP) (which is the same thing as FIP except that it calculates the number of home runs you should have allowed given the number of fly balls you allowed), and a league average HR/FB% are used. Since we aren’t using those parameters for WAR, we are hoping that we can estimate a player’s WAR using his statistics from the past five years. It is possible to get very accurate results if we include “situational” statistics that give information about when the various events happened. For instance, if we were to add “batting average with runners in scoring position,” this increases the accuracy of our estimates quite a bit. But you wouldn’t necessarily increase your statistic’s usefulness because we want to give a general overview of a player, not just for certain situations. This sort of sabermetric statistic may be more useful if a team were looking for players to sign before the postseason—when a player’s situational experience could be crucial for the team.

2.3 Literature Review

Even before the 2002 Oakland A’s brought moneyball to fame, baseball clubs had been using statistics to evaluate players. However, it wasn’t until the 2002 A’s that we saw a renaissance in baseball evaluation. On top of first basemen, general managers, and groundskeepers, baseball clubs now all hire statistical analysts. The vibrant community of sabermetrics has many methodologies and discussions on the best way to project every detail of baseball—such as home runs in cold weather months, or what factors impact the length of World Series games. One of the productions calculated the likelihood of a ball to land for a hit, based on the exit velocity and launch angle of the batted bat [3]. This is a meaningful question in baseball, since one of the most important things for a hitter is to put the ball in play. If a hitter strikes out then the play is over. However, if he places the ball in play there is a chance that the ball may drop in area without a defender, or that a defender commits an error and the batter reaches a base. This project shares some overlap with ours, as it uses copious amounts of data from baseball. Also, they are calculating a probability for a hit based off of specific inputs such as exit velocity and launch angle, while we are examining a player’s offensive worth (for the batting portion of the project) through standard baseball hitting statistics. The next project we looked at was presented at an analytics conference by a team from MIT that wanted to predict the next pitch type for a batter [4]. If a batter can correctly anticipate the next pitch type, he is in a better position to hit it. That is why pitchers worry about having their signs stolen or becoming too predictable in their pitch selection. This predictor incorporates information that is available to a batter such as the count, the current game state, the pitcher’s tendency to throw a particular type of pitch, etc. They used a linear support vector machine with soft-margin to build a separate predictor for each pitcher, and use the weights of the linear classifier to interpret the importance of each feature. They even compared their classifier to a naive classifiers and saw more success with their implementation than the naive classifier. Although this was a more concentrated analysis than most sabermetric statistics, it still provides a crucial prediction. It was interesting to see their algorithm against another algorithm that we covered in this semester of Machine Learning (Bayesian learning). The final project that we looked at was a much simpler but extremely similar project to our own. The project utilized clustering algorithms (KNN) to identify players who are similar to successful players in terms of process [5]. In practical terms, the objective is to find players who have struggled but hold the most potential relative to their process and are prime targets for acquisition. All of these projects were related to our project in that they used basic statistics to calculate an important prediction.

3 Results and Discussion

3.1 Solution

We had to have two types of testing. One for the pitchers and another for the position players. In each testing, we created instances that held the statistics as attributes for the player. For each player we kept their statistics from the last five years. For the labels in our prediction we will use WAR. We chose neural networks (NN), k-nearest neighbors (KNN), and ridge regression (RR) as the algorithms to feed our data into. The neural networks give back predication for a player’s WAR, while the KNN algorithm finds the k nearest neighbors and use and average of those neighbors WARs to calculate a probable WAR. Finally the RR would make general cross validations with the predicted WARs. All the preprocessing that was necessary to transform the raw-unorganized data was through our code. We used the scikit module in order to set up the implementations. In order to utilize the scikit KNN scikit NN scikit RR functions [6], we had to transform our python arrays into numpy arrays.

3.2 Experimental Setup

To evaluate our predictions we compared the predicted WARs with the actual WARs of a player. If the difference between the two WARs were underneath the threshold, we would declare that the program predicted the WAR correctly. The threshold we chose was 0.5. The reason we are not requiring our prediction to be exact is because we are dealing with continuous data, which makes it difficult to get exact predictions. The performance measurement we used was a percentage of the total number of players we predicted over the entire set of players. For KNN we tested our set with 5, 10, and 15 neighbors. For NN we test using two layers and varying nodes of 25, 50, and 100 neurons. With naïve Bayesian learning we used alphas of 0.01, 1x10-6 and 1x10-11. The alphas control the regularization. Regularization improves the conditioning of the problem and reduces the variance of the estimates, although with our data set it turned out to not make a difference.

3.3 Results

The machine learning algorithm that had the lowest percent error for both pitchers and batters was Bayesian learning. This makes sense given the data comes from human actions which are inherently probabilistic. The smallest error for batters was 0.5% error and for pitchers was around 1.25% error. KNN worked better overall than neural networks for predicting batting stats, however they were more similar when predicting pitching stats. This means that two batters with similar stats in one season will have more similar stats in the next than two pitchers in the same scenario which have a more complex relationship. The results for all the algorithms can be seen below in Figures 3 and 4. Our results show that there is a probabilistic relationship between similar player’s stats and how they will do in the next season. This makes sense given that a probabilistic relationship leaves room for players to have break out seasons or see their stats decline in the coming season, which happen often in sports.

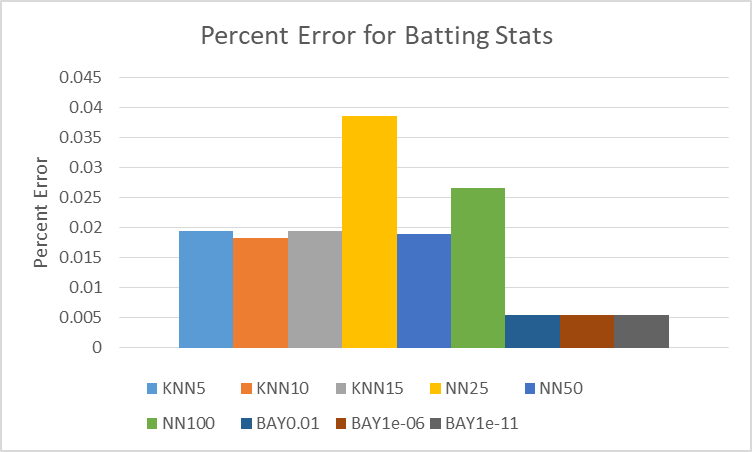
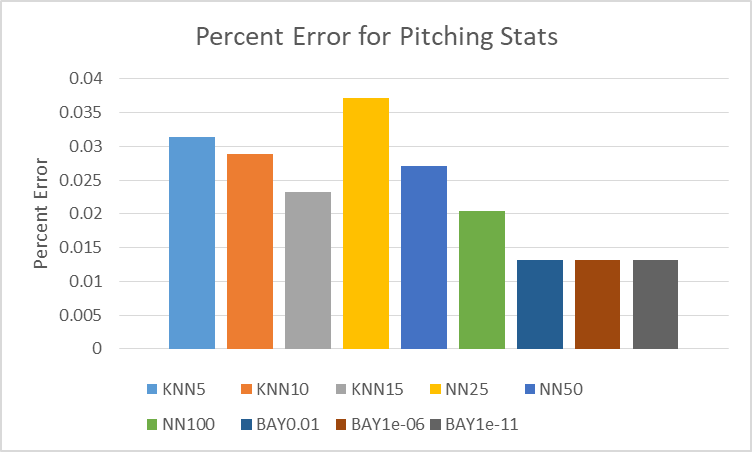
Figure 3: The percent error in batting stats for the machine learning algorithms.

Figure 4: The percent error in pitching stats for the machine learning algorithms.

4 CONCLUSIONS

Our love of sports and computer science has brought us to this project. Exploring sabermetrics gave us an insight into how statistics can give an entirely different perspective on sports. This project allowed us to implement another method of sabermetrics by using some of the algorithms we learned this semester. Its challenges included data wrangling and the implementation of modules that we had no prior experience with. However, those adversities made us understand machine learning more and gave us experience on learning how to use modules from documentation. Our results showed that Bayesian learning was the best in predicting a players future stats. This was likely due to the probabilistic nature of how well a player performs on any given day and in any given year.

We also explored some possibilities for this project in how we could expand it to different sports and expand the number of statistics that we use as attributes. The advancement of sabermetrics will continue to produce diverse, useful statistics that bolster the importance of analytics in baseball. This project is one of those instance that will continue to innovate and propel the use of statistics in baseball. The sabermetric community is searching for new methods to see if there is a novel path to producing the most competitive team possible through statistical analysis. Whether successful or not, methods like ours will continue to be worked on until there are further breakthroughs.

A.1 Introduction

A.2 Problem

A.2.1 Brief Introduction to Sabermetrics

A.2.2 Description of Data

A.2.3 Literature Review

A.3 Results and Discussion

A.3.1 Solution

A.3.2 Experimental Setup

A.3.3 Results

A.4 Conclusions

A.5 References

REFERENCES

|  |  |
| --- | --- |
| [1] | Birnbaum, P. (2012, June). A Guide to Sabermetric Research. Retrieved December 10, 2017, from http://sabr.org/sabermetrics |
| [2] | Baseball Reference. (2017, November). MLB Stats, Scores, History, & Records. Retrieved December 10, 2017, from https://www.baseball-reference.com/ |
| [3] | Albert, J. (2017, November 20). Exploring Baseball Data with R on WordPress.com. Retrieved December 10, 2017, from https://baseballwithr.wordpress.com/page/4/ |
| [4] | Ganeshapillai, G. (2012, March 2). Predicting the Next Pitch. Retrieved December 10, 2017, from http://www.sloansportsconference.com/content/predicting-the-next-pitch/. |
| [5] | Ishii, T. (2016, December). Using Machine Learning Algorithms to Identify Undervalued Baseball Players. Retrieved December 10, 2017, from <https://pdfs.semanticscholar.org/a8de/26ddaf197846890ef5f7a344de1b073f331e.pdf> |
| [6] | Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., & Mueller, A. (2012). Scikit API. Retrieved December 10, 2017, from http://scikit-learn.org/stable/index.html |