Use of Neural Networks to Predict the Win-Loss Percentage of Major League Baseball Teams

Nolan Donley
Department of Software Engineering
Mercer University
Macon, GA United States
nolan.s.donley@live.mercer.edu

***Abstract*— Major League Baseball (MLB) is the premier professional baseball organization and over the past few decades, baseball analytics have made a leap to the forefront of data science. Many attempts have been made in predicting the win-loss outcomes of specific matchups (between players or teams), but few have had relative success. In this article, we develop an artificial neural network (ANN) with one hundred and ten years of MLB regular season team statistics to predict a team's Win-Loss percentage (W-L%). The overall incentive for this model is to be used as an evaluation function for a genetic algorithm to determine best possible combinations of players for a team's total salary, desired play styles, etc. We aggregated data on the value of a team's batters and pitchers, and the results indicated that the model achieved high prediction accuracy given a team's data from the season. We were able to reach a mean squared error of 0.0014, which when squared (0.03742) is less than the testing set W-L% standard deviation of 0.07917. We also were able to get a mean game prediction difference (MGPD), a metric created by us to translate our results into an accuracy, of 4.7 which means the model can predict the number of games won for a team in a particular season to +/- 4.7 games. Future research may consider including traditional metrics or reducing the features set all together to make the prediction of hypothetical teams more feasible.**

Keywords—ANN Regression, major league baseball, prediction model, sabermetrics

I. Introduction

Major League Baseball (MLB) is the premier professional baseball organization and oldest major professional sports league in the world. Over the past few decades, due to the expansion of sports betting and the popularization of advanced statistical analysis, baseball analytics has experienced tremendous growth (1). Many teams are investing millions of dollars to develop their own departments to focus on this analysis and help advise in future decisions for the team. With baseball being a well-documented pastime, an abundance of data is available to the public with statistics for just about every feature of the game. Even so, analysis implementing machine learning (ML) techniques are still in their infancy. ML allows teams and other stakeholders to glean insights that are not readily apparent from human analysis (1).

Even with the growing number of academic articles in the space, we were unable to source any literature on the prediction of a team's win-loss percentage (W-L%) based on statistics from their season. Many attempts have been made in predicting the win-loss outcomes of specific matchups (between players or teams), but not in analysis of a team's potential. Because this research, to the best of our knowledge, appears to be primary, we will discuss a few practical applications. An important application, and our main incentive, is for an evaluation model of an arbitrary list of players. That is, given a list of 26 players, and given that those 26 players constitute an acceptable team composition, how good would the team be? We would like to use this as an evaluation function for a genetic algorithm to determine best possible combinations of players for a team's total salary, desired play styles, among other factors. This model could also be used to determine the teams with the best chance to advance through the playoffs. We will address this further in our discussion section.

In this article, we develop an artificial neural network (ANN) with one hundred and ten years of MLB regular season team statistics obtained from publicly available sources. The goal of our study is to evaluate the capability of an ANN regressor to accurately predict the W-L% for an MLB team using strictly advanced statistics.

We begin in Section 2 with background information on baseball analytics to help establish the context of this paper. In Section 3, we describe the methods we use for our data accumulation and preprocessing. We proceed in Section 4 by explaining our model architecture and training strategy. We then present our results in Section 5 and analysis in Section 6. Finally, Section 7 we conclude and provide some applicable ideas for future work.

II. Background

Statistics are the backbone of baseball analytics and can help a coach, manager, or owner determine a players' performance and worth. Many publications have been developed on the subject and given the MLB's revenue generation, it follows that performance prediction is an important and desired capability. Although the ML applications are known to these organizations, they are not often referred to in mainstream literature and culture (2). Simple statistics such as batting average or earned run average have been the main area of focus prior to the popularization of advanced statistics (sabermetrics) in the 1980s. Bill James is credited with popularizing the usage of sabermetrics, although no precise definition for the term exists. In practice, sabermetrics refers to any statistical analysis beyond the basic descriptive statistics (3).

An example of a sabermetric is a player's Power Speed Number (PSN) developed by Bill James (eq. 1). This is a simple formula computing the harmonic mean of a player's home runs (HR) and stolen bases (SB).

$$PSN = \frac{2*HR*SB}{HR+SB} \quad \text{(eq. 1)}$$

This number may help to explain a player's or performance but would be a poor predictor of W-L% as there is no variable for pitching or fielding. In our study, we use metrics such as this in combination with each other to predict a teams win percentage.

III. Methods

This research consisted of three stages (fig. 1) for development of the model.

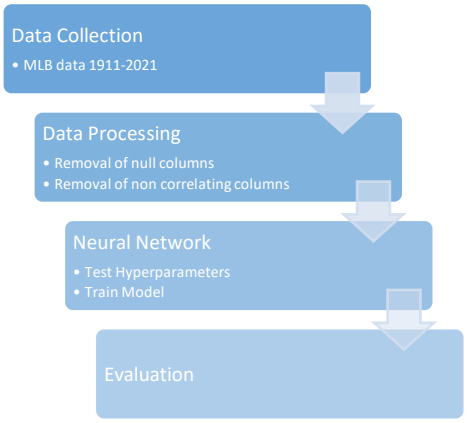

Fig. 1. Research Process

The aim of this study was to predict the W-L% of MLB teams at the end of a season given their value and sabermetric stats from the end of the same season, and this was achieved using deep learning and machine learning methods. First, we aggregated the season data of all teams from 1911 through 2021. Specifically, we gathered all value batting and value pitching statistics as well as sabermetric batting statistics from www.baseball-reference.com. Table 1 shows the statistics for one team for one season (Atlanta Braves, 2019). Next, we sanitized and normalized the data using Keras Normalization layer and then performed a principal component analysis with a Scikit-learn PCA decomposition to reduce the feature set (K. Team, S.-L Team). We then constructed a feed forward, deep learning regressor with an input layer, two hidden layers, and a linear output layer (fig. 2). For evaluation, we used mean squared error to determine the prediction accuracy of the model.

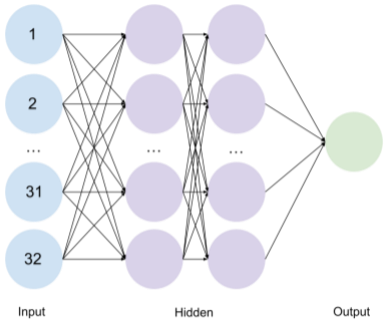

Fig. 2. Model Architecture

TABLE I. Atlanta Braves 2021 Statistics

Value Batting		Sabermetric Batting		Value Pitching	
PA	6056	**R/G**	4.91	**IP**	1410.2
Rbat	-47	**Outs**	4231	**G**	742
Rbaser	-1	**RC**	801	**GS**	161
Rdp	4	**RC/G**	5	**R**	656
Rfield	5	**AIR**	108	**RA9**	4.19
Rpos	56	**BAbip**	0.288	**RA9opp**	4.59
RAA	17	**BA**	0.244	**RA9def**	0.27
WAA	-0.2	**lgBA**	0.256	**RA9role**	0.01
Rrep	194	**OBP**	0.319	**RA9extras**	0.07
RAR	211	**lgOBP**	0.335	**PPFp**	104.8
WAR	19.1	**SLG**	0.435	**RA9avg**	4.61
waaWL%	0.501	**lgSLG**	0.433	**RAA**	71
162WL%	0.503	**OPS**	0.754	**WAA**	8.2
oWAR	19.2	**lgOPS**	0.768	**gmLI**	1.22
dWAR	0	**OPS+**	96	**WAAadj**	-0.7
		BtRuns	-41.9	**RAR**	202
		BtWins	-4.4	**waaWL%**	0.511
		TotA	0.724	**162WL%**	0.503
		SecA	0.301		
		ISO	0.191		
		PwrSpd	94.6		

[a.] Source: Baseball-Refernce.com. Please see webpage for abbreviation dictionary

A. *Model Architecture*

ANNs are some of the most used models for motion prediction in machine learning. A basic ANN comprises an input layer, a hidden layer, and an output layer. The number of hidden layers depend on the complexity of the problem. The higher the number of hidden layers, the slower the learning rate. In this study, because we are predicting a continuous value, we create a regression model.

We constructed an ANN prediction model in Python with an input layer of 32 neurons. The number of neurons in the two dense hidden layers were also set to 32 and use the rectified linear (ReLU) activation function. The output layer was a single neuron with a linear activation function. We used Keras-Tuner to determine the optimal number of neurons for the input and hidden layers as well as the number of hidden layers. We also we able to tune the optimizer function, kernel initializer, and activation functions for each of the layers. The Adam optimizer was employed with a learning rate of 0.001. The loss function was set to mean squared error, and the batch size was set to 64, and the epochs were set to 100.

IV. Results

We used TensorFlow and Keras to construct the deep learning model and the following sections present the prediction results obtained from our ANN regressor. Our data was split into training, validation, and testing sets with 80% going to training and 20% to testing. 20% of the training data was split for validation during training. One metric we have created for the intuitive digestion of our results is the mean game prediction difference (MGPD) (2). This is the average absolute difference in correct game prediction for a team's season. For example, if Team A wins 100 games out of 162, that gives them a W-L% of 0.617 and we would predict them to win 100 +/- MGPD games.

$$MGPD = G * \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad \text{(eq. 2)}$$

We were able to achieve a MSE of 0.00139, and MGPD of approximately 4.7 (2). Figure 3 shows our model's loss and validation loss as a function of epochs, and we also have included a few sample rows from the testing set to showcase a few predictions from our model (Table 3).

TABLE II. Evaluation Metrics

Metric	Value
Mean Squared Error	0. 00138769
Median Absolute Error	0. 02329364
W-L % Standard Deviation	0. 07911602
Mean Game Prediction Difference	4. 68578703

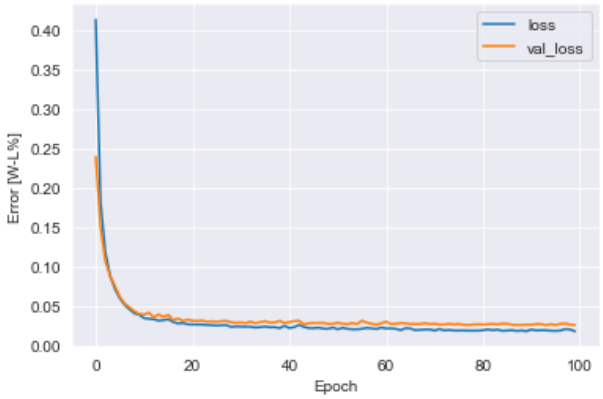

Fig. 3. Loss and Validation Loss as a function of Epochs

TABLE III. Actual vs. Predicted W-L% for MLB Teams

Team (Season)	Expected	Predicted
Boston Red Sox (2021)	0.568	0.551
Florida Marlins (1998)	0.494	0.501
Kansas City Monarchs (1948)	0.660	0.627
Cleveland Indians (2000)	0.556	0.562
New York Yankees (1977)	0.617	0.540
Atlanta Braves (1969)	0.574	0.559
New York Yankees (1936)	0.667	0.615
Pittsburgh Rebels (1915)	0.562	0.563
Texas Rangers (2006)	0.494	0.527
Kansas City Athletics (1963)	0.451	0.438
San Francisco Giants (1969)	0.556	0.552
New York Mets (1992)	0.444	0.434
Birmingham Black Barons (1943)	0.568	0.557
Philadelphia Phillies (1933)	0.395	0.426
Montreal Expos (1990)	0.525	0.502

V. Analysis

First, we analyze the formula for root mean squared error (RMSE) (eq. 5). By predicting the mean at every instance (eq. 6) we derive the standard deviation formula (eq. 7). Because the nature of this problem is so difficult, and our primary intention is to use this as a base indicator of hypothetical team performance in a genetic algorithm, we have declared any significant improvement above the W-L% standard deviation to be sufficiently effective. Taking the square root of our mean squared error of 0.0014 we get 0.03742 which is less than the testing set W-L% standard deviation of 0.07917.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2} \quad \text{(eq. 5)}$$

$$\hat{y}_1 = \hat{y}_2 \cdots \hat{y}_i = \mu \quad \text{(eq. 6)}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - \mu)^2} \quad \text{(eq. 7)}$$

In our study, we aggregated data on the value of a team's batters and pitchers. We used almost no traditional statistics and were able to achieve a high prediction accuracy. A major difference between our study and previous studies is the data used for constructing the prediction models. We collected value and sabermetric data from Baseball-Reference to establish whether these data points could accurately predict the team's W-L%. Complete game information on all pitchers would increase the accuracy of match outcome prediction. To the best of our knowledge, this is the first study to employ an ANN model to predict the W-L% of MLB teams, and the results indicate that the model achieved favorable prediction performance.

VI. Conclusions and Future Work

We predicted the W-L% of MLB teams by collecting the match data of all teams from the 1911 to 2021 seasons and using an ANN prediction model. The prediction accuracy of this model was evaluated. The prediction results indicated that the model achieved high prediction accuracy with a team's value and sabermetric data from the season. The prediction model proposed in this study achieved high prediction performance and can thus be used to provide some reference information for fans, team managers, and baseball enthusiasts. The proposed prediction models can also be used to predict game playoff outcomes since the model can basically predict who the best team should be, in contrast to the teams that have been experiencing some amount of luck.

Adjustments or modifications of the ANN model are encouraged to improve its prediction accuracy. In addition, we selected only value and sabermetric variables to develop the prediction models. Future research should consider including traditional metrics or reducing the metrics size all together. To use the model as an evaluator for a genetic algorithm, one needs to reliably be able to create all the input statistics from a team's individual player statistics. Another idea for future research would be to train a model to predict a teams win % based on the individual player statistics from its 26-man roster.

References

[1] Kaan Koseler & Matthew Stephan (2017) Machine Learning Applications in Baseball: A Systematic Literature Review, Applied Artificial Intelligence, 31:9-10, 745-763, DOI: 10.1080/08839514.2018.1442991

[2] Soto Valero,C.(2016).Predicting Win-Loss outcomes in MLB regular season games – A comparative study using data mining methods. International Journal of Computer Science in Sport,15(2) 91-112. https://doi.org/10.1515/ijcss-2016-0007

[3] Tolbert, B.; Trafalis, T. Predicting Major League Baseball Championship Winners through Data Mining. Athens J. Sport. 2016, 3, 239–252. https://www.athensjournals.gr/sports/2016-3-4-1-Tolbert.pdf

[4] Huang, M.-L.; Li, Y.-Z. Use of Machine Learning and Deep Learning to Predict the Outcomes of Major League Baseball Matches. Appl. Sci. 2021, 11, 4499. https://doi.org/ 10.3390/app11104499

[5] S.-L. Team, "Sklearn Decomposition PCA," Scikit-Learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

[6] K. Team, "Keras: Layer Normalization Layer," *Keras*. [Online]. Available: https://keras.io/api/layers/normalization_layers/layer_normalization/.

[7] J. Sobanski, "Fast and easy regression with Keras and tensorflow 2.3 ," *John Sobanski*, 28-Nov-2020. [Online]. Available: https://john.soban.ski/fast-and-easy-regression-with-tensorflow-part-2.html.