Comparison of Machine Learning Models to Best Predict Game Attendance in Major League Baseball

Seohyun Park

Abstract – To forecast Major League Baseball game attendance, this study employs six different regression models commonly used for machine learning. The models include Multiple Linear Regression, Decision Tree, Random Forest, Support Vector Regression, Multi-Layer Perceptron Regression, and Gradient Boosting. For the dataset, the 2022 and 2023 games of the Los Angeles Dodgers are used to reflect the recent trend of fan attendance and focus on a single random team so that different tendencies of each team's fans do not become a confounding variable. Each regression model's performance is evaluated by Mean Absolute Percentage Error and Root Mean Square Error. The performance evaluation suggests that Random Forest and Gradient Boosting predict attendance with the highest accuracy. Using these prediction models, each baseball team may facilitate staff management, event organization, and marketing before their games.

Keywords: Attendance, Machine learning, Major League Baseball, Mean Absolute Percentage Error, Regression models

1. Introduction

Each year during Major League Baseball (MLB) season, concerns about fan injuries always become an issue as foul balls cause serious wounds [1]. Some ballparks extend their netting to prevent such accidents, but others insist not to place nets to provide fans with a more vivid experience while watching games [2]. Therefore, for such ballparks where nets could be better prepared, staff deployment is necessary to ensure fans pay attention to foul balls. In this context, predicting attendance allows better staff deployment: placing security personnel in adequate spots effectively manages audience order and safety. Not only for audience safety, but attendance prediction is also helpful for event organizing and marketing, which are significant to team popularity and administration. If a very accurate attendance prediction is possible, people working in sports economics and marketing fields can make fruitful decisions for their teams.

However, accurate attendance prediction of baseball games tends to be more difficult than that of other sports, such as basketball or football, according to the study of Şahin and Uçar [3]. Despite this difficulty, this study attempts to forecast the number of audiences in each baseball game, using machine learning models commonly used for prediction. Common regression models that are employed for sports attendance forecasting appear in multiple studies: Şahin and Uçar identify Gradient Boosting (GB) as one of the machine learning methods that outperform the other methods, but they also prove that Artificial Neural Network (ANN) and deep Convolutional Neural Network are productive, similar to Park, Kim, Jeong, and Ahn who utilize Deep Neural Network (DNN) to predict the number of spectators in Korean Baseball League (KBO) [3,4]. Park also evaluates the performance of ANN for attendance prediction of KBO, which provides high-accuracy results [5]. With this prior knowledge, this study also makes use of GB and Multi-Layer Perceptron (MLP) regressor—a model based on ANN—, but it also employs four more machine learning methods: Multiple Linear Regression (MLR), Decision Tree (DT), Random Forest (RF), and Support Vector Regressor (SVR). By fitting these models to game data, this study compares their performances and identifies which model best forecasts MLB game attendance.

Unlike previous studies, the present study utilizes recently updated games for the dataset, which will reflect attendance trends for the 2022 and 2023 seasons.

2. BACKGROUND

2.1 Machine Learning Models

2.1.1 Gradient Boosting (GB)

GB is an ensemble machine learning method that is practical as it corrects errors of the previous models. The ensemble machine is a method of constructing and combining multiple base estimators to create one final model, allowing the weighted sum of the weak learners to become a strong model. Starting with a weak classifier at the leftmost decision tree in Figure 1, GB adds a tree for each iteration, and the new learner contains the previous classifier's prediction residuals that tell how much the expected value differs from the actual value. By repeating this process, models approach a final strong classifier.

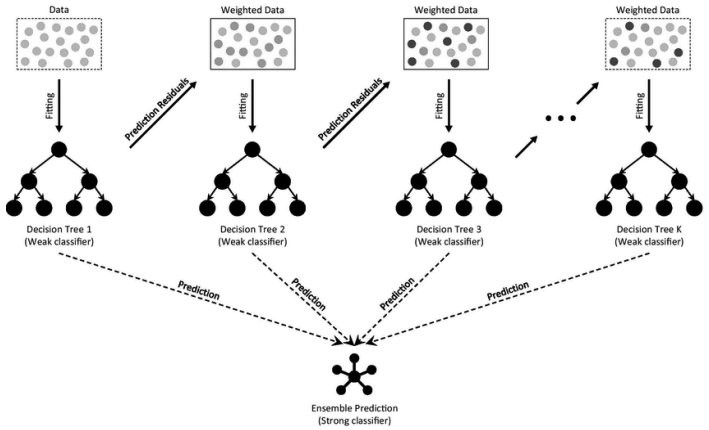

Figure 1. The Architecture of Gradient Boosting Decision Tree [7]

Due to this feature, GB is highly robust to overfitting: overfitting is a situation in which a model accurately predicts the data used for training but has poor prediction accuracy for test data that is not used in the learning process. In other words, GB has high generalizability, and it means that a large number of data leads to better performance.

2.1.2 Multiple Linear Regression (MLR)

MLR utilizes multiple explanatory variables to predict the outcome with the formula below:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon \quad (1)$$

where y_i and x_i are dependent and independent variables respectively, β_0 is a y-intercept, β_p is slope coefficients for each explanatory variable, and ϵ is the model's error term. Since this model assumes that a linear relationship between dependent and independent variables exists, it may not capture intricate patterns, if any exist.

2.1.3 Decision Tree (DT)

DT manages various independent variables and produces an outcome. The DT algorithm begins at the root node, as shown at the top of Figure 2. Each node of DT is evaluated against the training dataset, and it has multiple branches, which are represented as arrows in Figure 2. Leaf nodes are the output of the decision node, and they do not split into additional sub-nodes.

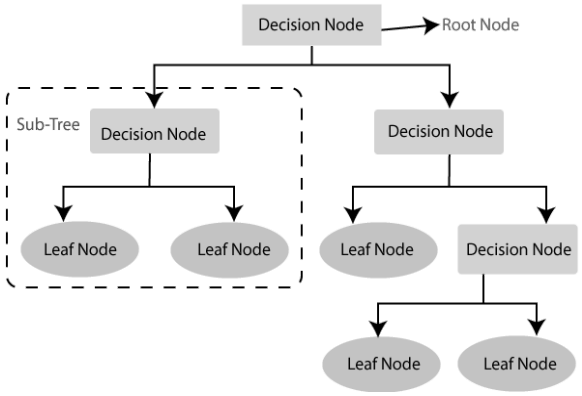

Figure 2. The Architecture of Decision Tree
(https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)

2.1.4 Random Forest (RF)

Similar to GB, RF uses an ensemble learning method for regression. Each tree in the model is built by drawing a random sample from the test sample input. During the hyperparameter tuning, the number of these trees is indicated by n_estimators. The trees in RF run in parallel, so trees do not interact with each other while building them. Then, predictions from each tree are averaged to produce an outcome, as shown in Figure 3.

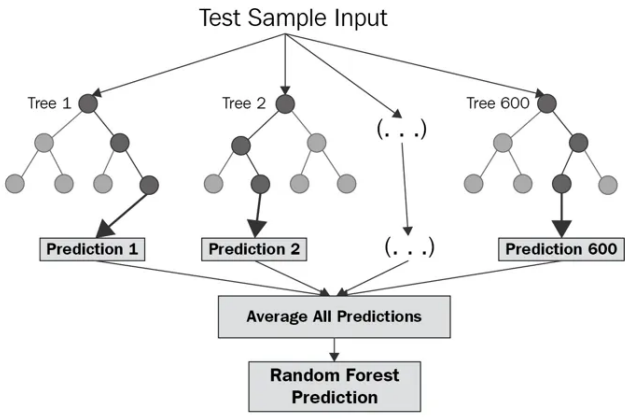

Figure 3. The Architecture of Random Forest Regressor
(https://corporatefinanceinstitute.com/resources/data-science/random-forest/)

Besides, each RF tree finds the best node split from a random subset, and its size can be given by a hyperparameter called max_featuers. These two sources of randomness are a strength of RF since it is

efficacious in preventing overfitting.

2.1.5 Support Vector Regressor (SVR)

SVR finds a hyperplane that best analyzes the relationship between the input variable (x_n) and target value ($\hat{y}$), so it is effective in high dimensional spaces. It uses a subset of training points that influence the position and orientation of the hyperplane, and these data points are called support vectors, which are represented as $K(x_n,x)$ in Figure 4. The bias term is presented as b.

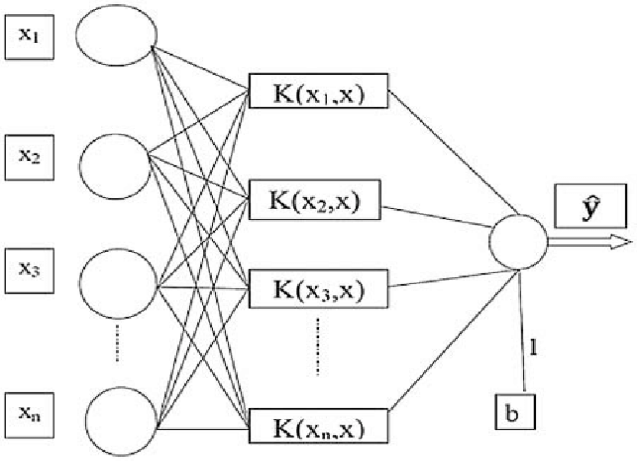

Figure 4. The Architecture of SVR [8]

SVR can detect various types of relationships in the data by using different kernel functions, but it may struggle to analyze datasets containing outliers or noisy data.

2.1.6 Multi-Layer Perceptron (MLP) Regressor

MLP Regressor is a type of ANN, a machine learning method with performance and characteristics similar to the human brain. This regressor can capture complex non-linear relationships in data with multiple hidden layers, as shown in Figure 5. Each hidden layer can learn other layers' characteristics and gain more intricate patterns from lower layers that are relatively simpler. Each layer has multiple neurons that transform the input data (X_n) through a series of weighted connections ($w_{n,n}$) and nodes (u_n), and weights for the neurons are adjusted during training using backpropagation.

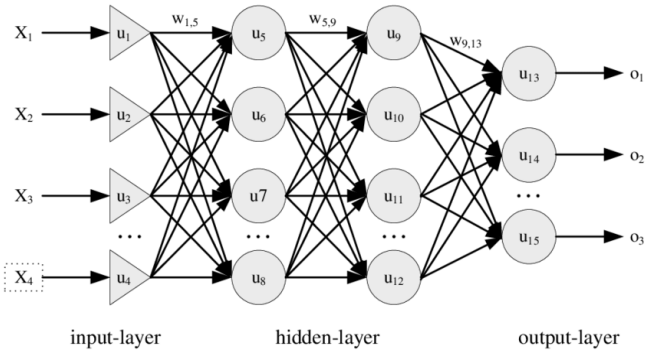

Figure 5. The Architecture of MLP Regressor [9]

However, as the hidden layers deepen, the model becomes vulnerable to overfitting. Thus, careful regularization is often needed to overcome this issue. During the hyperparameter tuning process, the regularization can be done by assigning a value to a parameter named alpha that represents the strength of the L2 Regularization term.

3. Methodology

3.1 Dataset

The data includes the Dodgers games in 2022 and 2023, which gives 324 games in total. Among 30 MLB teams, the Los Angeles Dodgers were chosen by a simple random sampling method. The data is team-specific because this study does not allow various teams' different fan characteristics to become a confounding variable. Besides, the data only includes the last two years because this study aims to reflect the recent trend of fan attendance; however, 2020 and 2021 data are not usable as they significantly lack the audience due to the COVID-19 pandemic [10]. Attendance has been restored since 2022, so the last two years were selected for this study's dataset. All the game data was taken from a website named 'Baseball Reference' (https://www.baseball-reference.com), using Scrapy Spiders in Python. Among the 324 games, one game had a null attendance value, so it was replaced by the mean attendance value of the 324 games.

The data contains nine input variables and one dependent variable, as shown in Table 1. Since this study aims to forecast the number of audience, attendance per game is the target value. The input variables were determined after reviewing multiple studies of baseball attendance prediction: features most likely to affect fan attendance were chosen [3-6]. The features can be classified into time, climate, and game characteristics. Among the categories, time and climate are outside-game factors, and game characteristics are inside-game factors. An input variable 'Weather' is a condition right before a game starts, and it consists of four types of weather: 'sunny,' 'cloudy,' 'overcast,' and 'in dome.' From the game characteristics, 'Home or Away' indicates the Los Angeles Dodgers; it is a boolean variable, meaning the input is either True or False. Another game characteristic, the Championship Leverage Index, represents the importance of the game on the team's probability of winning the World Series. 1.0 is average importance, so any value greater than one means the game is more important than the average, and vice versa. Dodgers Winning Streak counts the Los Angeles Dodgers' latest consecutive winning or losing games. Winning streaks are expressed as positive integers, and losing streaks are expressed as negative integers.

Table 1. Input and output variables.

Input		Output
Time	Month	Attendance per game
	Day of Week	
	Start Time	
Climate	Temperature	

	Weather
Game Characteristics	Opponent
	Home or Away
	Championship Leverage Index
	Dodgers Winning Streak

Among the input variables, categorical features include 'Month,' 'Day of Week,' 'Weather,' 'Opponent,' and 'Home or Away.' All these categorical features were preprocessed by one-hot encoding because label encoding may cause a machine learning model to apply priority to a certain categorical value, which is unwanted. For example, when encoding the 'Opponent' feature, there are 29 opponent MLB teams, but if one team is labeled 0 while another team is labeled 28, then a model may consider the team labeled 28 as a more critical factor. One-hot encoding effectively prevents this situation, so all categorical values were preprocessed this way.

3.2 Model Construction

Before fitting models to data, the dataset was split into a train set and a test set, using the 'train_test_split' scikit-learn module. While splitting, the test_size hyperparameter—which represents the proportion of the test data—was set to 0.2, and the random_state hyperparameter was set to 10 for reproducibility. Therefore, training data had 259 games, and testing data had 65 games. Since nine independent variables were used for prediction, the X train's dimension was 259 by nine, and the X test's dimension was 65 by nine. In order to fit each model to this split dataset, the scikit-learn package imported the machine learning methods. Hyperparameters of each model were determined after conducting ten trials; hyperparameters that yielded the best score were chosen. For DT, max_depth was 10, but for RF, max_depth was set to 15, and n_estimators was set to 200. When fitting the MLP Regressor, hidden_layer_sizes was ten thousand, and alpha was set to 0.001.

3.3 Evaluation

Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) were employed to evaluate the performance of six different models. Both calculations were imported by metrics in scikit-learn. MAPE represents the ratio of errors that differ from each observed value on average when comparing actual observed values and predicted values as follows in Equation (2):

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (2)$$

where A_t is an actual value, F_t is a forecast value, and n is the number of fitted points. MAPE always gives a non-negative float, and zero percent is the best possible value, as zero percent means no difference between the actual and predicted values. RMSE calculates how far predictions fall from measured actual values using Euclidean distance as follows in Equation (3):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}}, \quad (3)$$

where y(i) represents an actual observation, $\hat{y}$(i) represents a corresponding prediction, and N is the number of data points. Like MAPE, zero is the best value for RMSE, but the unit of RMSE is not in percentage but in people. For instance, in this study, an RMSE value of 5000 means the actual and predicted number of audience are different by 5000 people.

4. Result and Discussion

When comparing the MAPE and RMSE values of six different models, RF and GB show the best performance in prediction. As shown in Table 2, the RF regressor proves its effectiveness in forecasting MLB attendance, with only 12.003% off from the actual game attendance. Similar to the study of Şahin and Uçar, GB also shows very high accuracy, with only a 12.279% difference [3]. Both RF and GB have RMSE values of around 5000, confirming that their performances surpass the other machine learning methods.

Table 2. MAPE and RMSE of each regressor

	GB	MLR	DT	RF	SVR	MLP
MAPE (%)	12.279	19.952	14.390	12.003	26.958	25.522
RMSE	5203	7568	7467	5337	9963	9856

After RF and GB, DT and MLR also provide compelling prediction accuracy, with MAPE values of 14.390% and 19.952% respectively. However, Figure 6 reveals that SVR and MLP are not accurate enough for game attendance prediction, producing the highest MAPE and RMSE values. Their RMSE values almost approach 10,000; considering that the capacity of Dodger Stadium is 56,000 seats, 10,000 is almost one-fifth of the stadium. Consequently, the two models with poor prediction require further improvement for administrators and marketers to exploit.

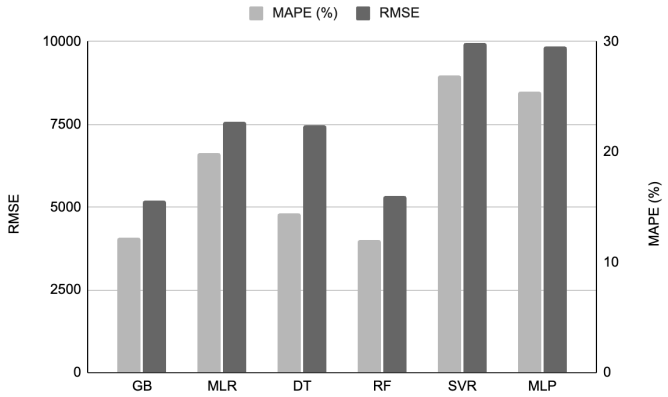

Figure 6. Bar graphs of MAPE and RMSE

Since the best two models—RF and GB—are both ensemble methods, this result reveals that an ensemble machine works the best when predicting MLB game attendance. Considering the ensemble method's features, the machine's high generalizability and robustness over a single estimator seem to influence this result because numerous factors determine MLB game attendance, so even when two games have the same condition, it cannot be assured that the two games will have the same number of audience. Therefore, an ability to combat overfitting is essential for these predictions. In that sense, it is understandable that MLP Regressor produces low accuracy here since one of the biggest problems of ANN-based models is overfitting.

5. Conclusion

With data from the 2022 and 2023 games of the Los Angeles Dodgers, this study attempted to correctly forecast MLB fan attendance per game. Unlike other studies, various machine learning models were compared, and nine input variables that are considered to affect attendance were utilized. The result found that RF and GB are most effective in predicting MLB game fan attendance among six different machine learning models. A reasonable inference that explains this outcome would be the ensemble method's trait that prevents overfitting: the randomness of RF especially helps overcome this issue. When MAPE was calculated, the most accurate result was 12.003%, indicating that predictions were very successful. However, a limitation of this study is the small amount of data. Since the RF regressor performs better with a large dataset, the model might have given a more accurate forecast if more game data had been included. As this study only focused on a single team's schedule and attendance, getting predictions from other teams' perspectives and combining results may also help improve the model's performance. In addition, although hyperparameters that were thought to be optimal were selected after a specified number of trials, since better scores may be obtained, superior performance can be expected in the future through more specialized tuning methods.

Using this fan attendance prediction model, baseball team operation may be more systemized: knowing how many tickets will be sold helps manage a budget, more suitable and successful events can be organized based on the number of spectators, and better marketing strategies can be developed by further analyzing which factors most significantly influence the fan attendance. Even for staff deployment, the prediction model is helpful, and accidents in ballparks can be prevented, which promotes more fun and safe experiences for baseball fans.

References

[1] Milsten, A., Bradley, W. F., Hill, M., Sacco, W., & Henes, M. (2022). Foul ball rates and injuries at Major League Baseball games: a retrospective analysis of data from three stadiums. Prehospital and Disaster Medicine, 37(2), 277-283.

[2] Tak, M., Nguyen, V., Enoch, J., & Lehren, A.W. (October 1, 2019). Foul balls hurt hundreds of fans at MLB ballparks. See where your team stands on netting. Retrieved from https://www.nbcnews.com/news/sports/we-re-going-need-bigger-net-foul-balls-hurt-hundreds-n1060291

[3] Şahin, M., & Uçar, M. (2022). Prediction of sports attendance: A comparative analysis. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology, 236(2), 106-123.

[4] Park, D.J., Kim, B.W., Jeong, Y.S., & Ahn, C.W.. (2018). Deep Neural Network Based Prediction of Daily Spectators for Korean Baseball League: Focused on Gwangju-KIA Champions Field. Smart Media Journal, 7(1), 1-8.

[5] Park, J., & Park, S. (2017). A Study on Prediction of Attendance in Korean Baseball League Using Artificial Neural Network. KIPS Transactions on Software and Data Engineering, 6(12), 565-572.

[6] Mueller, S. Q. (2020). Pre-and within-season attendance forecasting in Major League Baseball: a random forest approach. Applied Economics, 52(41), 4512-4528.

[7] Deng, Haowen & Zhou, Youyou & Wang, Lin & Zhang, Cheng. (2021). Ensemble learning for the early prediction of neonatal jaundice with genetic features. BMC Medical Informatics and Decision Making. 21. 10.1186/s12911-021-01701-9.

[8] Thomas, Sonia & Pillai, G.N. & Pal, Kirat. (2016). Prediction of peak ground acceleration using ϵ-SVR, ν-SVR and Ls-SVR algorithm. Geomatics, Natural Hazards and Risk. 8. 1-17. 10.1080/19475705.2016.1176604.

[9] Permanasari, A., Chamsudin, A., & Wahyunggoro, O. (2013). Utilization of Neural Network for Disease Forecasting. Proceedings 59th ISI World Statistics Congress.

[10] Gough, C. (November 15, 2023). Major League Baseball: total attendance at regular season games from 2006 to 2023. Retrieved from https://www.statista.com/statistics/193421/regular-season-attendance-in-the-mlb-since-2006/