# A Deep Learning Approach to Predict Home Win-Loss in Major League Baseball Season Games

*Wei, Quek (n10503196)*

*IFN703 Assessment 3, due 11:59pm Friday 30 October 2020*

## Executive Summary

It is fair to say that the Major League Baseball (MLB) has entered into a new phase ever since the book "Moneyball: The Art of Winning an Unfair Game" was published in 2003 [1]. Nowadays, it is no longer uncommon but rather a necessity for a baseball team to hire a group of full-time data analysts, also known as "*Sabermetricians*", as part of the front office team. They analyse game statistics and players performance; and use the analysis result to augment game decision making that give the team a winning edge. With an average MLB team worth $1.78 billion in 2019 [2], predicting the winner of an MLB season is therefore important to team owners to ensure they are getting value for their investment money. By using publicly available MLB data, this study aims to explore the suitability of implementing a deep-learning-based machine learning model to predict MLB season game outcomes (home win or loss). Accumulative performance statistics of both home and visiting teams were calculated from past game data prior to a match. By calculating Pearson's correlation coefficients between features, some of these statistics were removed to reduce dimensionality. A Neural Network model was fitted, and its performance for predicting game outcomes (home win or loss) of MLB season games was assessed. The result of a model trained on 3-seasons data prior to the predicted season showed a mean of 55.6% prediction accuracy for each season over a decade (2010-2019), 1.76% better than the baseline (home teams win all games), and an >60% accuracy achieved on the most recent season (2019).

## Introduction

Americans are passionate about baseball. Major League Baseball (MLB), a popular professional baseball league that runs its regular season from April through to October, is now worth billions of dollars. MLB receives approximately $2 billion per year from media rights contract [3], and $1.1 billion dollar from legal sports betting [4]. Winning games is therefore essential because it attract higher television deals, higher ticket sales, higher licencing and sponsorships, and more importantly, higher revenues.

To predict the outcomes of baseball games, many have attempted to seek for solutions quantitatively. "*Sabermetrics*", a popular term used among the baseball analytics community, is emerging to be a critical part of MLB [5]. Due to the vast amount of publicly available recorded game data covering almost all aspects of a baseball game, MLB lends itself well to data analytics. Analysts can now apply machine learning models to ingest these large baseball datasets in order to elicit meaningful insights about team and player performance.

In this study, a deep-learning-based machine learning model will be fitted to explore its accuracy on predicting the result of each MLB game. By using the dataset available from *Retrosheet*, different metrics from historical data will be used to fit this neural network model. However, baseball data are quite *noisy* – with approximately 200,000 different events in each season and over 164 different features in each event. *Is it possible to predict a baseball game outcome using a deep-learning machine learning approach*?

## Literature Review

Almost everyone who is interested in sports covets on winning a competition. After all, as described by a well-known quote in sports – "Winning isn't everything; it's the only thing". Major League Baseball (MLB) games are known of its competitiveness. The best team wins only 60% of the games played and the worst team 35% [6]. Yet, many are still interested in developing systems with the aim of predicting game winners accurately using the sheer volume of baseball data available [7].

Many MLB teams are now embracing data analytics. Teams have invested heavily to setup their own analytics team [8]. This phenomenon has been well-documented by many authors [5, 7]. However, though machine learning is well understood by the teams' analytics team, knowledge remains an off-limits to public access. Baseball analytics is a competitive industry and most analysts and teams choose to hold their work proprietary, if not, behind a paywall.

Whilst there are never short of machine learning related research papers that focusing on identifying salary-efficient players, very limited studies have been done on predicting the winner of a baseball game. Still, several papers are found to be useful to this project [9, 10]. For example, Yang and Swartz [9] calculated the probability of a team winning a certain game by using a Bayesian approach. They used past performance data, batting ability, starting pitcher and home field advantage to create a model to predict winners of each game using a two-stage Bayesian model. However, it should be note that, their work was to provide prediction of division ranking, at several points in time during a regular season only (May 30, June 30, July 30 and August 30).

In another study, Soto Valero [10] performed a comparison of different machine learning methods (kNN, artificial neural network, Decision Tress and SVM), and managed to achieve a prediction accuracy of just 60% with SVM. However, as he had acknowledged in their paper, the result was not "particularly remarkable", even by introducing some of the advanced sabermetrics numbers.

Due to the large number of baseball statistical numbers available, it is therefore important to identify which of these statistics are strong determinants of winning or losing a game. To do this, *Sabermetrics* that have the most impact on runs scored are identified. Many of these have been evaluated in the past studies [6, 9, 10] and the following predictors are specifically found to be useful for this study: (for more detailed descriptions, see Appendix B)

| Battling | Pitching | Team Stats |
|---|---|---|
| On-Base Percentage (obp) | Earned Run Average (era) | Plate appearance (pa) |
| Slugging Percentage (slg) | Hits allowed (h) | Number of errors (e) |
| Batting Average (ba) | Bases Allowed (b) | Left-on base (lob) |

| | | |
|---|---|---|
| Runs Batted In (rbi) | Strike-outs (k) | Win-Loss record (w_prct) |
| Walk Drawn (wb) | Walk Thrown (wp) | Head-to-head Winning (h2h_prct) |
| Runs Scored per game (rg) | Innings Pitched (ip) | Log5 (log5) |
| On-Base Plus Slugging (ops) | Walks plus hits (whip) | Pythagorean Expectation (pythE) |
| Batting Average on Balls in Play (babip) | Hits Per 9 Innings (h9) | Win-Loss as home/away (w_prct_home/away) |
| | Home Runs per 9 Innings (hr9) | Win-Loss record as day/night game (day_night) |
| | Walks per 9 Innings (bb9) | Day Rest before match [1] (day_rest) |
| | Runs allowed per game (rag) | |

**Table 1. Selected Baseball Statistics**

Various studies have indicated that deep-learning models are useful for predictions in sports domain [11, 12]. However there has been a lack of research in predicting MLB game outcome using deep-learning approach. Baseball statistics data are often broad, complex and non-linear in nature which are better suited for a deep-learning model. A neural network could be promising because of its capacity of modelling high dimensionality, multitude combinations dataset with any given function.

In summary, this literature review has identified that there is a need to conduct further research to provide evidence that a deep-learning machine learning model is a viable approach to predict the winner of an MLB game. These literature review has also provided a basis for formulating needed features to train a deep-learning model for game outcome prediction.

## Approach

Figure 1 summarised the approach applied in this study:

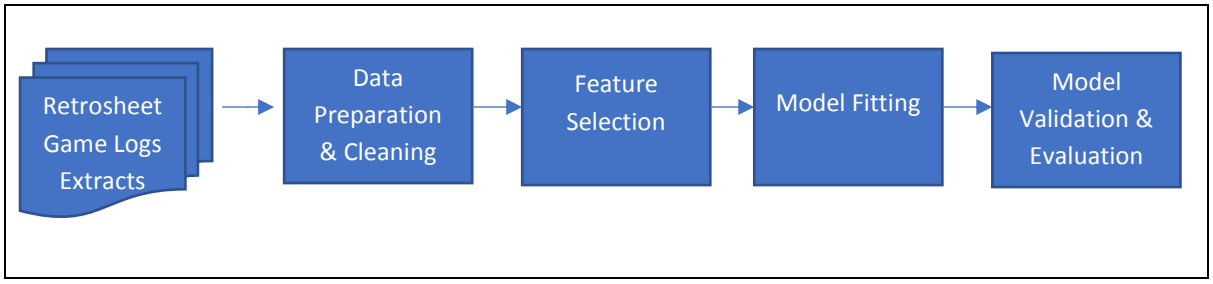


**Figure 1. A graphical representation summarising the approach applied in this study**

### Data Gathering

Historic baseball datasets are publicly available from various online sources in different formats. In this study, the primary source of baseball data was obtained from *Retrosheet*. The dataset contains extensive detailed game logs of all MLB games in each season, with records dated as far back as the late 1800s, though older data are less detailed and less accurate. The availability of large amount of past season data makes it an optimal and rich dataset for fitting machine learning models. Often machine learning models require sufficient training data to achieve high accuracy. Past game log

data for machine learning is useful as the game outcome is already included in the dataset. This allows the model's predictive accuracy to be assessed during testing.

For this study, game logs from seasons 2004 to 2019 were downloaded. There were 162 games for each of the 30 MLB teams in each season, which equal to 2,430 games played per season. In total there were about 39,000 game log records. Each game log record contains 161 columns about the game such as game date, venue, attendance, pitching and batting statistics, players, managers and umpire crews, etc., of both home and visiting team. Appendix A contains descriptions of all columns in *Retrosheet* game logs.

## Data Preparation and Cleaning

Collected data are pre-processed before used. Dataset are checked for consistency, cleaned, and formatted appropriately. Firstly, downloaded game logs extracts are loaded chronologically into a single dataframe as Retrosheet only provides one extract per season. Next, unused attributes are removed. About half of the attributes are descriptive data which are not useful for calculating team performance metrics.

### Data Completeness

The completeness and integrity of the data has a direct influence the predictive performance of a machine learning model. To assess the quality of the data gathered and loaded, the column `'acquisition_info'` is been examined. According to the data dictionary supplied by Retrosheet, it indicates the completeness of the game data. All game logs downloaded have the full complete game logs data as indicated.

### Duplicate and Missing Data Check

Duplication check has also been performed to ensure no duplicate game data exist in the downloaded or data mistakenly loaded more than once. Missing data are also checked. There are 4 variables containing null values which are mostly info columns. Due to high percentage of missing values (>80%), these variables are removed from the dataset.

### Other Data Quality Check

Several reasonable data checks have been performed such as checking the number of games played by each team per season, identifying any outliers, etc. This is to ensure the final dataset is of quality and ready for the next stage. During the checking process, a mismatch was found on the MLB team code between 2011 and 2012. It was caused by the fact that MLB Team Florida Marlins (FLO) has been renamed to Miami Marlins (MIA) in 2012 after moving its home park from the suburb of Miami Gardens to the Miami city.

### New Attributes

Two new attributes are added into the dataset in preparing dataset for machine learning: `'season'`, which represents the MLB season the game belonged to, and `'home win'`, which is a binary flag indicating whether the game is won by the home team or not.

## Machine Learning

### Calculating Metrics

For predicting the result of a game correctly, it is important that any information about the game predicted are excluded. Both teams' accumulative performance metrics of all past games are calculated prior to the game to be predicted. The basic idea is to use these statistics data to learn about team performance progression and thereby predicting the game outcome using a machine learning algorithm. This statistical-based understanding of how teams progresses over time is then used to make inferences about the teams' performance in the upcoming game and how different metrics might affect the game outcome. Figure 2 is a graphical representation of the four accumulated statistics calculated for Oakland Athletics and San Francisco Giants during MLB Season 2019.

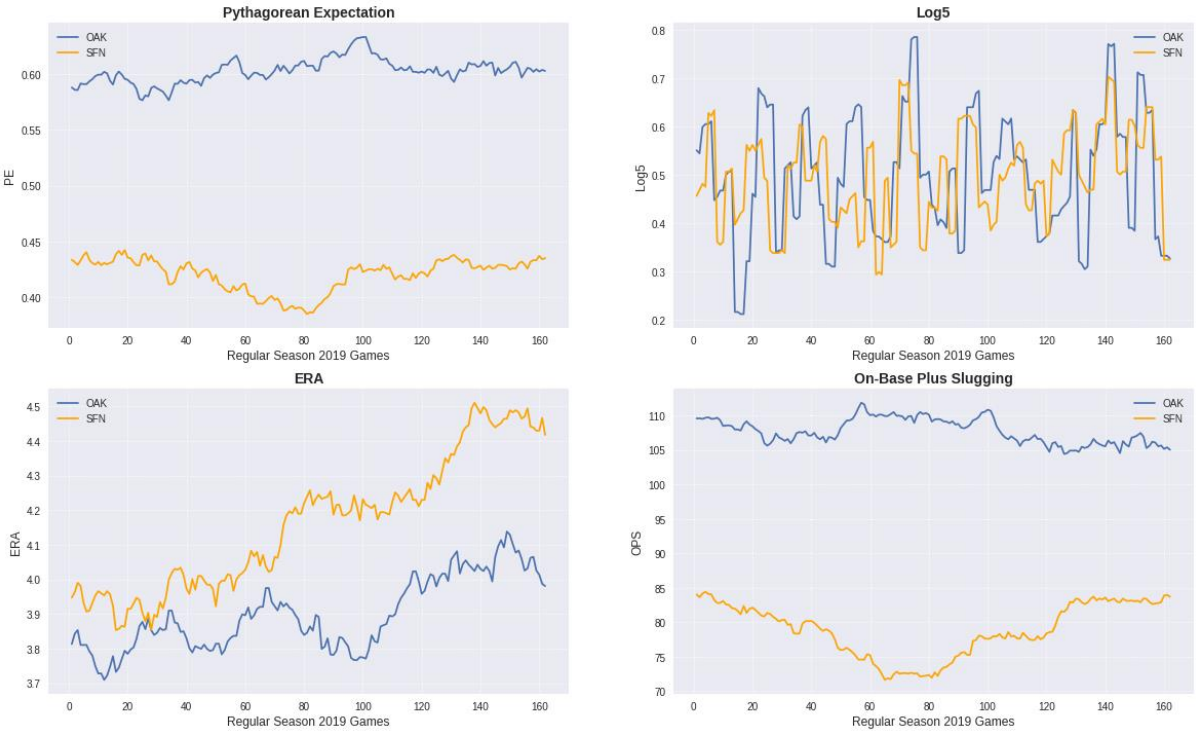


**Figure 2. Selected Baseball Statistics**

### Assumptions

There are several assumptions when selecting and calculating these features specific to this study. Specifically, (1) every statistics are derived at team level, not at player level; (2) player factors, such as injury, suspensions, trades and health issues, though might be available prior the match, we have simplified this calculation process by excluding them; (3) park factor is excluded; (4) seasonality and weather factors are not considered.

### Final Values

For every game record between 2005 to 2019 seasons, both home and visiting teams' performance metrics are calculated from the past 162 game logs, which is approximately the number of games a team plays in an MLB regular season. Looking at multiple years data might not be ideal as team form changes over time which impact performance. For every statistic, it represents how much the difference between the home team and the visiting team. For example:

*ERA = Home Team ERA – Visiting Team ERA*.

## Feature Selection

Due to the high dimensionality of the datasets, correlations between features are examined. It is important to determine which features have the higher influence on the probability of winning a game. Features that have a high correlation to each other are also removed in preparation for machine learning model. Before examining the correlations, every metric has been standardised such that its distribution has a mean value 0 and standard deviation of 1.

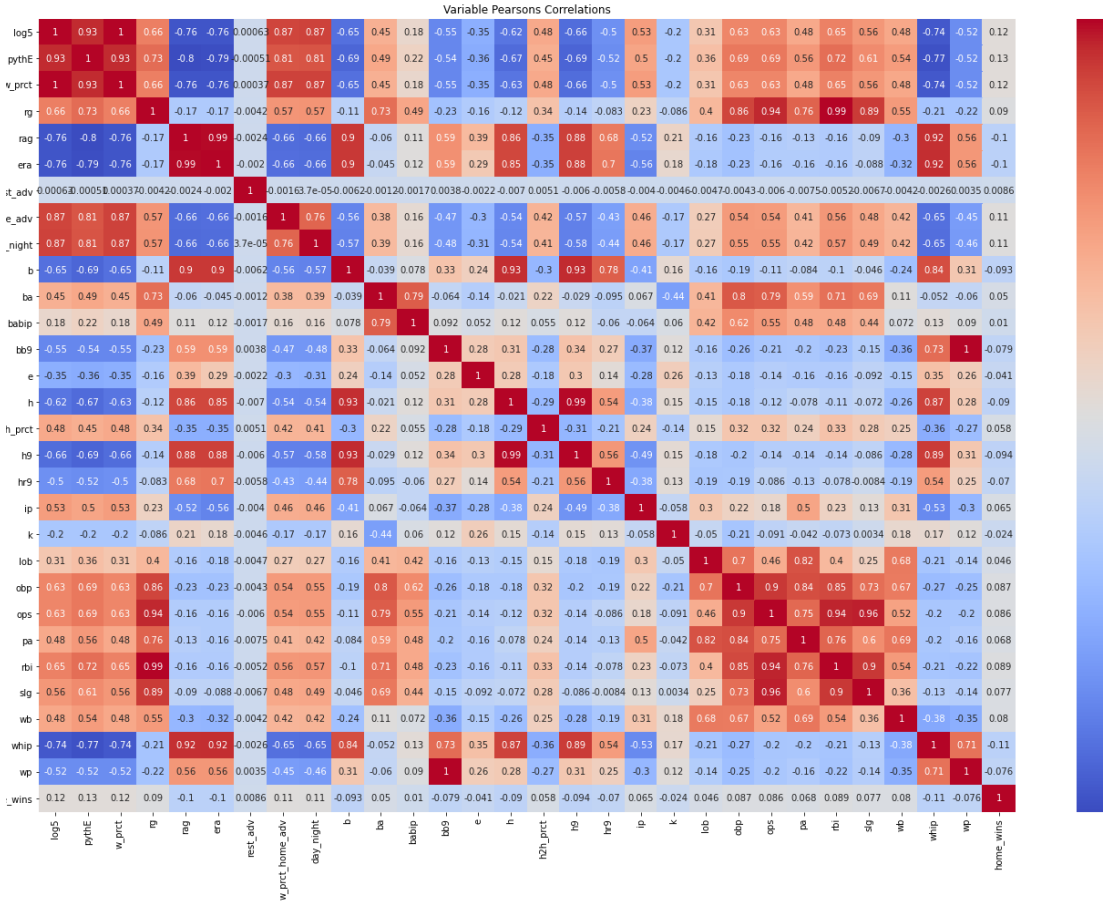


**Figure 3. Correlation Matrix before Feature Selection**

Several features were removed where there is a correlation of >0.8 and <-0.8 with another features. Selections were also based on how highly correlated they are to the response variable. After this elimination process, the dimensionality has reduced from 29 to 14. Multicollinearity is therefore been reduced in the dataset and would enable the machine learning model training to be more efficient and effective.

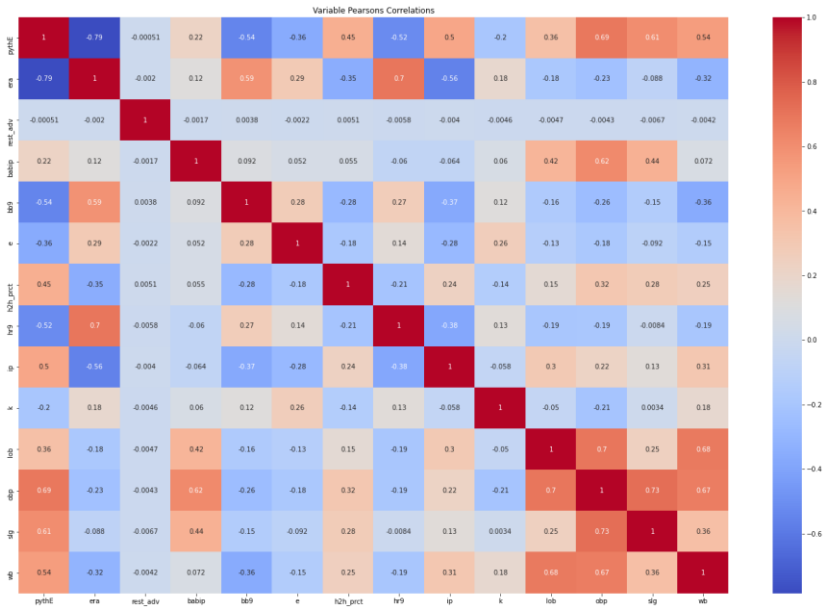


**Figure 4. Correlation Matrix after Feature Selection**

## Splitting Data

After the dataset is cleaned and treated, it is partitioned into different subset – training and test sets, to be ready for the next stage. By using a hold-out protocol, the dataset is split into training set and testing set. The testing set consists a full season game data, whereas the training set contains season game data prior to the testing set.

## Neural Network Model

A simple, feed-forward neural network is trained using binary cross entropy loss function as the deep-learning model. This is the most basic Neural Network architecture type commonly used for binary classification application. The first layer is the input layer, and the last layer is the output layer. Hidden layers are layers between input and output layers. A deep neural network is any network that has more than one hidden layer.

Several parameter values were tested using Grid Search during the model fitting process to determine the optimal values. A weight matrix was also calculated on each training set to address the class imbalance issue in the dataset.

The architecture of the Neural Network model fitted is shown in Figure 5. From the process of finding the optimal parameter values, 4 hidden layers with 16 nodes each are found to be the optimal. All the hidden layers use *ReLU* (Rectified Linear Units) as the activation function. *ReLU* function returns input directly if the value is positive, otherwise a zero value is returned. The output node uses a sigmoid activation function that transformed a vector into a value between 0 and 1 in a form of a sigmoid curve.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 14)                0
_________________________________________________________________
dense (Dense)                (None, 16)                240
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
```

```
_________________________________________________________________
dense_1 (Dense)              (None, 16)                272
_________________________________________________________________
dropout_1 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_2 (Dense)              (None, 16)                272
_________________________________________________________________
dropout_2 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_3 (Dense)              (None, 16)                272
_________________________________________________________________
dropout_3 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 34
=================================================================
Total params: 1,090
Trainable params: 1,090
Non-trainable params: 0
_________________________________________________________________
```

**Figure 5. The architecture of Neural Network model fitted**

For other parameters, the model uses *Adam* as its optimiser. The loss function used is *binary cross entropy*. It produces a logistic regression for binary classification in a sigmoid curve which representing the probability of a point belonging to the positive or negative class. Figure 6 shows an example of how the function works graphically. The green bar represents the probabilities predicting the true class, whereas the red bar is the probabilities of predicting the negative class

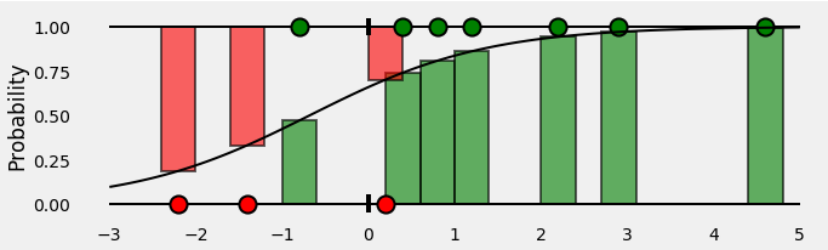


**Figure 6. Binary Cross Entropy Loss Function**

The model is configured to train for 100 epochs with a batch size of 20. An early stopping is also been configured in the model fitting process to stop training when a convergence is achieved after 10 iterations.

# Findings

## Data Exploration

### Percentage of games win as home vs visiting team by team and season

The home team has a clear advantage of winning the game, and it is consistently across all seasons, which is reflected on the left-hand side heatmap in Figure 7. Overall, the home vs visiting team winning percentage ratio is approximately 54:46, with only a few extreme exceptions such as HOU in 2015 (29.63%) and DET in 2019 (27.16%).

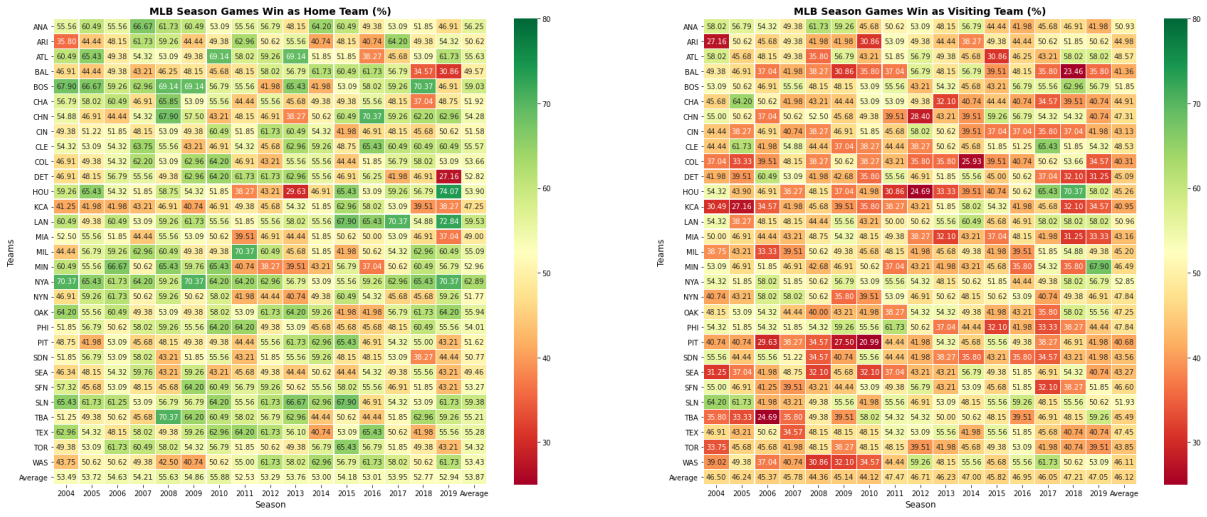


**Figure 7. Percentage of Home Wins Games by Season**

## Percentage of home wins between day and night games

On average home team has a winning advantage of games played during the day across all season, exception season 2006 (53.38% vs 55.23%), 2013 (52.93% vs 54.24%) and 2015 (53.83% vs 54.35%). This is despite that two-third of the games are played at night.

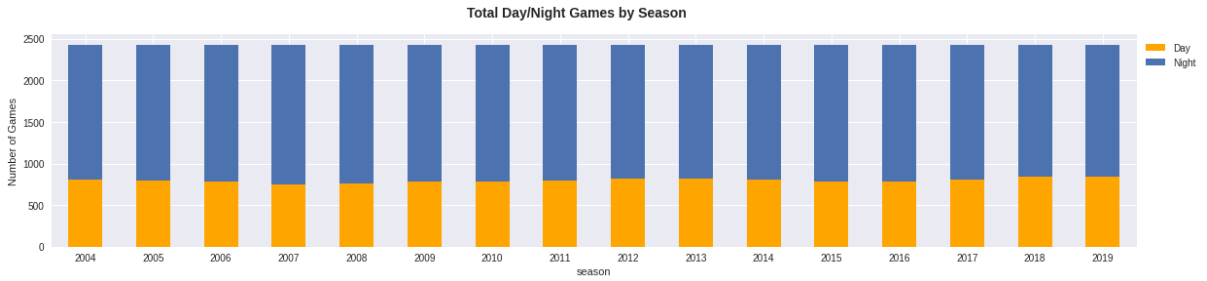


**Figure 8. Total day and night games per season**

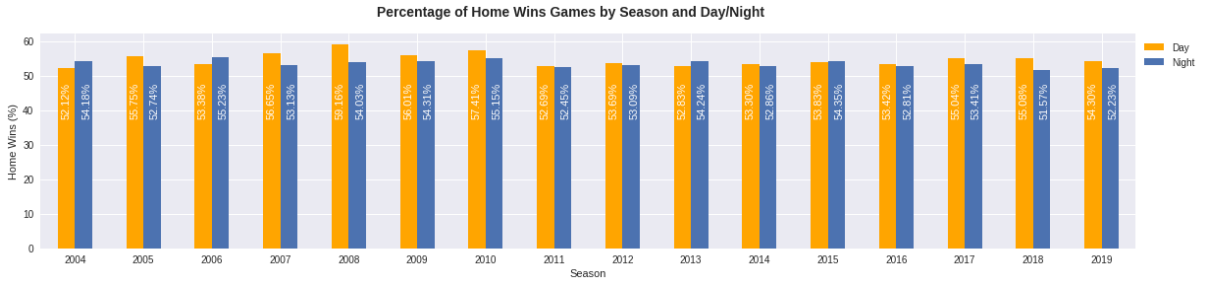


**Figure 9. Percentage of home wins games by season and day/night**

## Percentage of day or night games win by team and season

There are some differences in game winning percentage for team playing day and night matches. From Figure 10, MLB Teams performed more consistently in the night matches than day matches. Overall, despite the average winning percentage ratio is approximately 50:50 for 2004-2019 seasons, there are some notable extremes winning percentage for some team for day games.

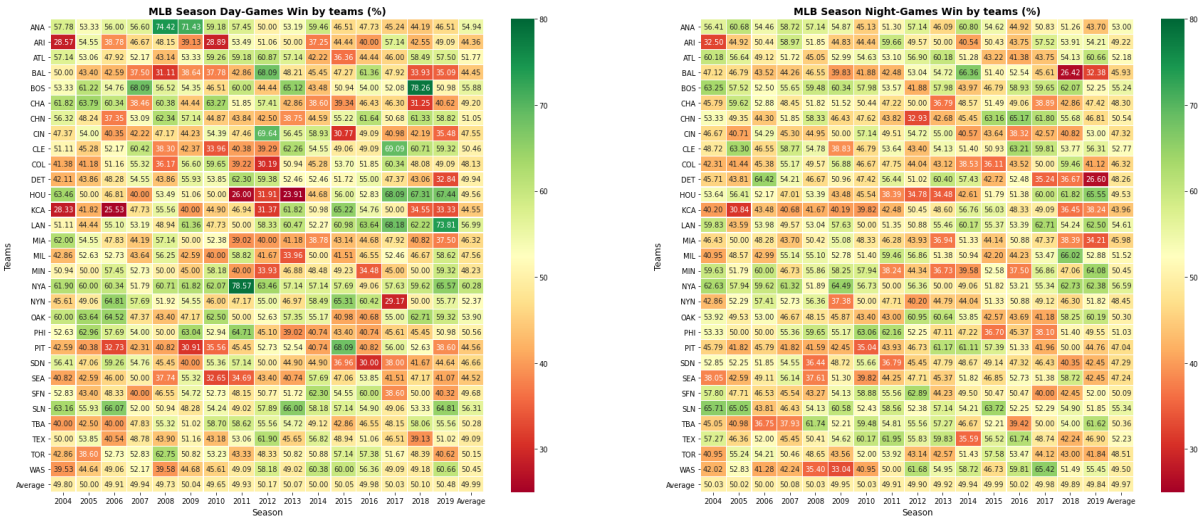


**Figure 10. Percentage of day/night games win by team and season**

## Model Performance Evaluation

The model presented here has been evaluated for its predictive accuracy using a hold-out protocol. It has been progressively tested to predict each season between 2010 and 2019, which is equivalent to running 10 testings using hold-out. Cross-validation is not suitable as it would result using model trained on future games to predict past games.

The dataset is split into training set and testing set. Each testing set consists a full season game data, whereas the training set only contains game data prior to the testing set.

The accuracy is defined as how accurate the model predicts the home team in winning or losing the game. This evaluation approach is consistent with most of the literature reviewed [9, 10], whereby this predictive accuracy measures the proportion of correctly predicted games.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Here, true positive (TP) means correct prediction of the home team wins; true negative (TN) means correct prediction of the home team losses; false positive (FP) means incorrect prediction of losses and wins and false negative (FN) means incorrect prediction of wins as losses.

The model was evaluated using past 3, 4 and 5 seasons of data that leading up to the next season. For example, using past 3-season data approach, to test using the 2019 season data, the model would be trained on 2016-2018 season data. This approach ensures the model is not trained on future season to predict the past season game outcome which would be meaningless in real-life context.

- Model 1: Trained on past **3** seasons
- Model 2: Trained on past **4** seasons
- Model 3: Trained on past **5** seasons

To compare the model prediction performance result, a baseline is defined for evaluation. It is simply predicting that home teams will win all games in all seasons. The performance of the model on testing set and the comparison to the baseline is summarised as followed:

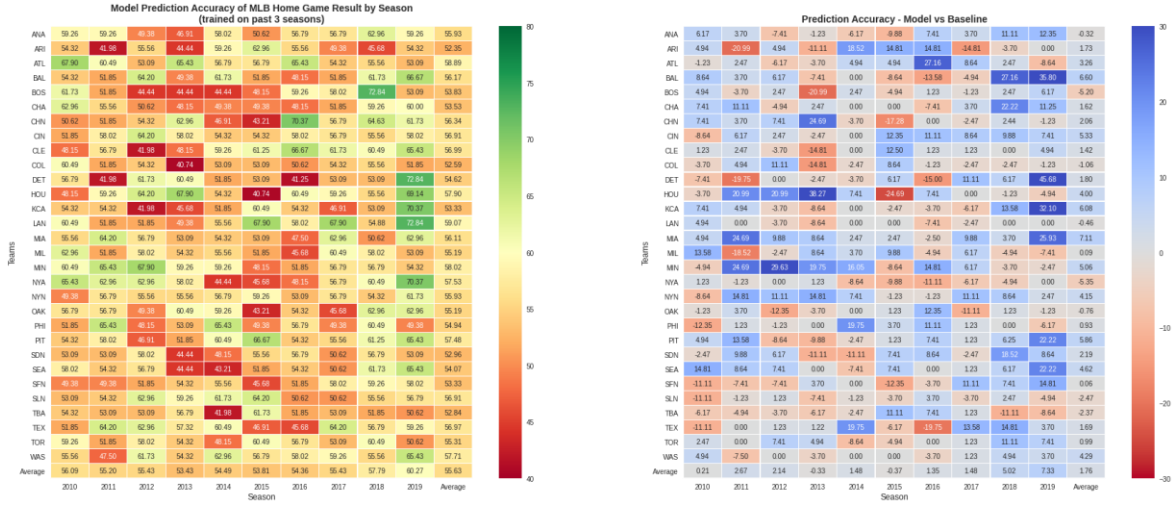


**Figure 11. Prediction accuracy of model trained on past 3 seasons (i.e. training set 75%, testing set 25%) and a comparison to the baseline**

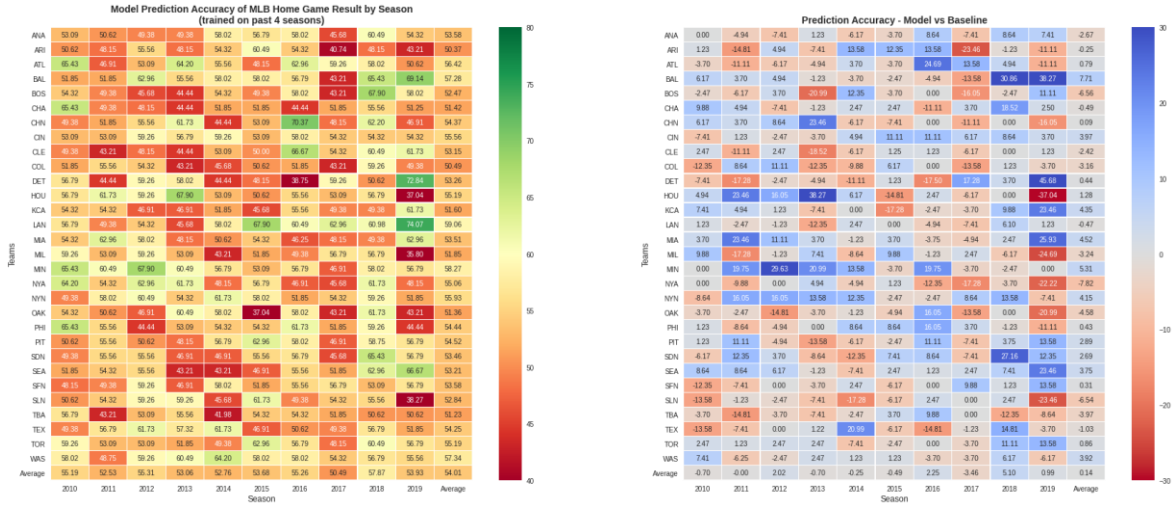


**Figure 12. Prediction accuracy of model on past 4 seasons (i.e. training set 80%, testing set 20%) and a comparison to the baseline**

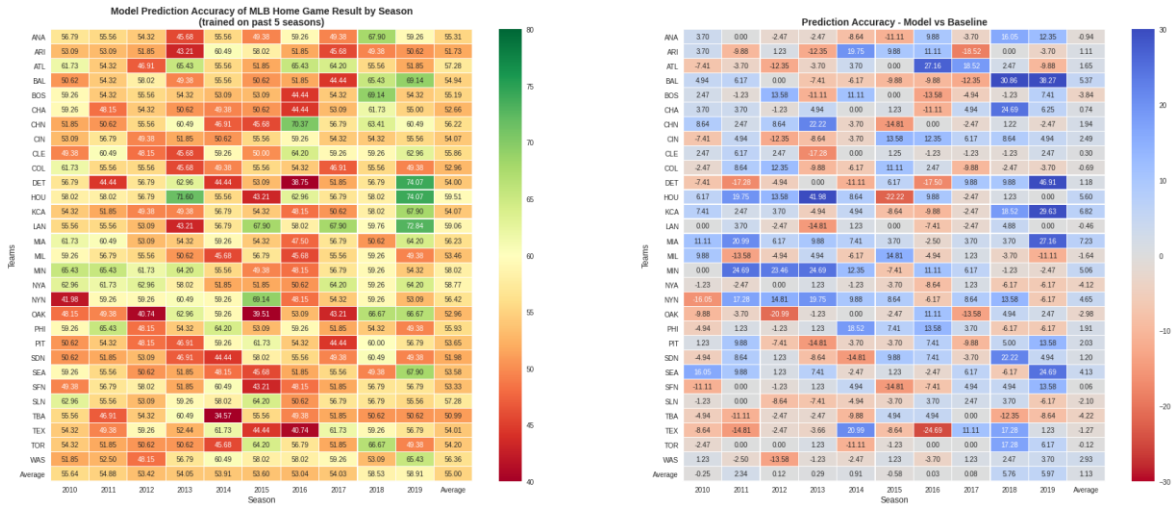


**Figure 13 Prediction accuracy of model on past 5 seasons (i.e. training set 83%, testing set 17%) and a comparison to the baseline**

| Season | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model 1 | 0.21 | 2.67 | 2.14 | -0.33 | 1.48 | -0.37 | 1.35 | 1.48 | 5.02 | 7.33 | 1.76 |
| Model 2 | -0.70 | 0.00 | 2.02 | -0.70 | -0.25 | -0.49 | 2.25 | -3.46 | 5.10 | 0.99 | 0.14 |
| Model 3 | -0.25 | 2.34 | 0.12 | 0.29 | 0.91 | -0.58 | 0.03 | 0.08 | 5.76 | 5.97 | 1.13 |

**Table 2. Summary of prediction accuracy (%) differences between models and baseline for season 2010-2019**

Figure 11, 12 and 13 show the prediction results of model 1, 2 and 3 across all teams in all season. The results of using different training dataset size are compared and found to be similar. There were some encouraging results in predicting most recent 2019 season: Model 1 has predicted 16 teams with over 60% accuracies, with 4 teams over 70% accuracy. However, an attempt to train the model with data augmentation technique (by increase the size of training dataset) failed to improve accuracy, as shown in the result of Model 2 and 3. In fact, the performance went worse as more older data are added.

All models have performed slightly better than the baseline, with Model 1 performs the best out of the 3 models. On average, the result is 1.76% better than the home advantage. Again, the result shows that models trained with more data has not improve the performance, which further suggest that there is less value in adding older game logs to the training dataset.

Also, it is worth noting that Model 1 is 7.33% and 5.02% better than the home advantage baseline for season 2019 and 2018 respectively. This indicates that deep-learning model is indeed able to learn the patterns and trends from the statistics used, especially for the most recent season games.

## Reflection

In this study, a great amount of time was spent on gathering and preparing the data and calculating metrics as features. Extra care has also been applied in researching and validating the calculated features against some of the data available online to ensure their accuracy.

Though the results are found to be only slightly better than flipping a coin, the model achieved an average of 1.76% for predicting season 2010-2019 better than the baseline that selects the home team to win every game. To improve the accuracy, one possible area to explore is to refine and expand the feature set used in this study. There are some features which have not be considered in this study due to time and computational constraints. For example, one could analyse data at player level, or include past performance statistics of the starting pitcher. External factors such as park factor, forecasted weather and player salaries could also be included as these might have an influence on predicting a game outcome. Still, these new features must be carefully assessed using the appropriate feature selection technique as many of these terms might not help the prediction than just add more noise to the data.

In addition, there is also other machine learning algorithms could be considered, such as K-Nearest Neighbours (KNN), Support Vector Machine (SVM), or clustering methods (e.g. K-Means clustering). The approach could also be modified from a simple binary classification to a regression model to predict the runs scored.

In conclusion, with the correct metrics selected, this study has demonstrated that it is feasible of using a deep-learning model to predict the outcome of an MLB game. This model could provide not

only for game outcome prediction, but also insights to strengths and weaknesses of teams from win perspectives, which could be beneficial to MLB managers.

## References


[1] M. Lewis, Moneyball: The Art of Winning an Unfair Game, New York: W.W. Norton & Company, 2003.

[2] Forbes, "Baseball Team Values 2019," 10 04 2019. [Online]. Available: https://www.forbes.com/sites/mikeozanian/2019/04/10/baseball-team-values-2019-yankees-lead-league-at-46-billion. [Accessed 13 08 2020].

[3] CNBC, "Major League Baseball's new media rights deal with Turner Sports worth over $3 billion," 15 06 2020. [Online]. Available: https://www.cnbc.com/2020/06/16/mlb-new-media-rights-deal-with-turner-sports-worth-over-3-billion.html. [Accessed 13 08 2020].

[4] American Gaming Association, "How Much Do Leagues Stand to Gain from Legal Sports Betting?," 18 10 2018. [Online]. Available: https://www.americangaming.org/resources/how-much-do-leagues-stand-to-gain-from-legal-sports-betting/. [Accessed 13 08 2020].

[5] G. B. Costa, M. R. Huber and J. T. Saccoman, Reasoning with Sabermetrics: Applying Statistical Science to Baseball's Tough Questions, Jefferson: McFarland & Company, Inc., Publishers, 2012.

[6] W. D. Kaigh, "Forecasting Baseball Games," *Chance,* vol. 8, no. 2, pp. 33-37, 1995.

[7] B. Baumer and A. Zimbalist, The sabermetric revolution: Assessing the growth of analytics in baseball, Philadelphia: University of Pennsylvania Press, 2014.

[8] M. Henshon, "Big Data Baseball: Math, Miracles, and the End of a 20-Year Losing Streak," *Scitech Lawyer,* vol. 12, no. 1, pp. 18-20, 2015.

[9] T. Y. Yang and T. Swartz, "A Two-Stage Bayesian Model for Predicting Winners in Major League Baseball," *Journal of Data Science,* vol. 2, pp. 61-73, 2004.

[10] C. Soto Valero, "Predicting Win-Loss outcomes in MLB regular," *International Journal of Computer Science in Sport season games – A comparative study using data mining methods,* vol. 15, no. 2, pp. 91-112, 2016.

[11] B. G. Aslan and M. M. Inceoglu, "A comparative study on neural network based," in *Seventh International Conference on*, 2007.

[12] W. A. Young, W. S. Holland and G. R. Weckman, "Determining Hall of Fame Status for Major League Baseball Using an Artificial Neural Network," *Journal of Quantitative Analysis in Sports,* vol. 4, no. 4, 2008.

[13] Retrosheet, "Retrosheet," [Online]. Available: www.retrosheet.org.

# Appendices

## A. Retrosheet Game Logs

Retrosheet Data Notice:

Recipients of Retrosheet data are free to make any desired use of the information, including (but not limited to) selling it, giving it away, or producing a commercial product based upon the data. Retrosheet has one requirement for any such transfer of data or product development, which is that the following statement must appear prominently:

```
The information used here was obtained free of
charge from and is copyrighted by Retrosheet.  Interested
parties may contact Retrosheet at "www.retrosheet.org".
```

Retrosheet makes no guarantees of accuracy for the information that is supplied. Much effort is expended to make our website as correct as possible, but Retrosheet shall not be held responsible for any consequences arising from the use the material presented here. All information is subject to corrections as additional data are received. We are grateful to anyone who discovers discrepancies and we appreciate learning of the details.

| Field(s) | Description |
|---|---|
| 1 | Date as a string in the form "yyyymmdd" |
| 2 | Number of the game corresponding to the current season. |
| 3 | Day of the week as a string. |
| 4-5 | Name and league of the visitor team. |
| 6 | Game number of the visitor team. |
| 7-8 | Name and league of the home team. |
| 9 | Game number of the home team. |
| 10-11 | Runs of the visitor and home team, respectively. |
| 12 | Length of game in outs. A full 9-inning game would have a 54 in this field. If the home team won without batting in the bottom of the ninth, this field would contain a 51. |
| 13 | Day/night indicator ("D" or "N"). |
| 14 | Completion information indicates if the game was completed at a later date (either due to a suspension or an upheld protest). |
| 15 | Forfeit information. |
| 16 | Protest information. |
| 17 | Park identifier. |
| 18 | Attendance. |
| 19 | Duration of the game (in minutes). |
| 20-21 | Visitor and home line scores as a string. For example, "010000(10)0x" indicates a game where the home team scored a run in the second inning, ten in the seventh and didn't bat in the bottom of the ninth. |
| 22-38 | Offensive statistics of the visitor team: at-bats, hits, doubles, triples, homeruns, RBI, sacrifice hits, sacrifice flies, hit-by-pitch, walks, intentional walks, strikeouts, stolen bases, caught stealing, grounded into double plays, awarded first on catcher's interference and left on base(in this order). |
| 39-43 | Pitching statistics of the visitor team: pitchers used, individual earned runs, team earned runs, wild pitches and balks (in this order). |

| | |
|---|---|
| 44-49 | Defensive statistics of the visitor team: putouts, assists, errors, passed balls, double plays and triple plays (in this order). |
| 50-66 | Offensive statistics of the home team. |
| 67-71 | Pitching statistics of the home team. |
| 72-77 | Defensive statistics of the home team. |
| 78-79 | Home plate umpire identifier and name. |
| 80-81 | First base umpire identifier and name. |
| 82-83 | Second base umpire identifier and name. |
| 84-85 | Third base umpire identifier and name. |
| 86-87 | Left field umpire identifier and name. |
| 88-89 | Right field umpire identifier and name. |
| 90-91 | Manager of the visitor team identifier and name. |
| 92-93 | Manager of the home team identifier and name. |
| 94-95 | Winning pitcher identifier and name. |
| 96-97 | Losing pitcher identifier and name. |
| 98-99 | Saving pitcher identifier and name. |
| 100-101 | Game Winning RBI batter identifier and name. |
| 102-103 | Visitor starting pitcher identifier and name. |
| 104-105 | Home starting pitcher identifier and name. |
| 106-132 | Visitor starting players identifier, name and defensive position, listed in the order (1-9) they appeared in the batting order. |
| 133-159 | Home starting players' identifier, name and defensive position listed in the order (1-9) they appeared in the batting order. |
| 160 | Additional information |
| 161 | Acquisition information |

## B. Features Definition

| Name | Description |
|---|---|
| On-Base Percentage (obp) | A measure of how often a batter reaches base. It is approximately equal to Times on Base/Plate appearances. |
| Slugging Percentage (slg) | The number of total bases divided by the number of at bats. |
| Batting Average (ba) | The number of hits gotten by a player divided by his number of at bats. |
| Runs Batted In (rbi) | A run batted in is credited to the batter for the number of runners who score due to any hit, fielder's choice, out, walk or HBP by the batter. Runs that score as the result of double plays (or the ultra rare bases loaded triple play) or errors do not result in credit being given for an RBI. |
| Walk Drawn (wb) | A walk or base on balls, abbreviated BB, occurs when a player gets on base by drawing four balls from the pitcher. A walk might be intentional. |
| Runs Scored per game (rg) | Average runs scored per game over a given period. |
| On-Base Plus Slugging (ops) | A player's overall offensive performance. OPS is the sum of on-base percentage and slugging percentage. |
| Batting Average on Balls in Play (babip) | Batting average on balls in play is a measure of the number of batted balls that safely fall in for a hit not including home runs |
| Earned Run Average (era) | A pitcher's Earned Run Average (aka ERA) is a primary measure of his success. It is expressed as an average number of opponents' earned runs scored per notional nine inning game |

| | |
|---|---|
| Hits allowed (h) | Hits Allowed is the statistic used to track the number of hits a pitcher gives up. |
| Bases Allowed (b) | Number of bases allowed to opponent. |
| Strike-outs (k) | An out called when a batter has made three strikes. |
| Walk Thrown (wp) | A walk or base on balls, abbreviated BB, occurs when a player gets on base by drawing four balls from the pitcher. A walk might be intentional. |
| Innings Pitched (ip) | Measures the length of a pitcher's appearance by outs. |
| Walks plus hits (whip) | WHIP stands for walks plus hits, all divided by innings pitched. It is one of the standard measures of a pitcher's efficacy, noting how well they prevent baserunners. |
| Hits Per 9 Innings (h9) | H/9 represents the average number of hits a pitcher allows per nine innings pitched. It is determined by dividing a pitcher's hits allowed by his innings pitched and multiplying that by nine. |
| Home Runs per 9 Innings (hr9) | HR/9 represents the average number of home runs allowed by a pitcher on a nine-inning scale. The statistic is determined by dividing a pitcher's home runs allowed by his total innings pitched and multiplying the result by nine. |
| Walks per 9 Innings (bb9) | Walks per nine innings tells us how many walks a given pitcher allows per nine innings pitched -- using the formula walks divided by innings times nine. |
| Runs allowed per game (rag) | Total runs allowed divided by number of games played |
| Plate appearance (pa) | A plate appearance refers to a batter's turn at the plate. Each completed turn batting is one plate appearance. |
| Number of errors (e) | A fielder is given an error if, in the judgment of the official scorer, he fails to convert an out on a play that an average fielder should have made. Fielders can also be given errors if they make a poor play that allows one or more runners to advance on the bases. |
| Left-on base (lob) | Refers to the number of men who remain on base at the end of an inning |
| Win-Loss record (w_prct) | The number of wins and losses that a team has accumulated over a given period |
| Head-to-head Winning (h2h_prct) | The percentage of winning between two teams over a given period or number of games |
| Log5 (log5) | Estimate the probability that team A will win a game, based on the true winning percentage of Team A and Team B. |
| Pythagorean Expectation (pythE) | Estimate the percentage of games a baseball team "should" have won based on the number of runs they scored and allowed. Comparing a team's actual and Pythagorean winning percentage can be used to make predictions and evaluate which teams are over-performing and under-performing. |
| Win-Loss as home/away (w_prct_home/away) | The percentage of winning as home/away team. |
| Win-Loss record as day/night game (day_night) | The percentage of winning game played in day/night game |
| Day Rest before match (day_rest) | Number of days between last played game and upcoming game. Maximum value is 3. |

## C. Codes