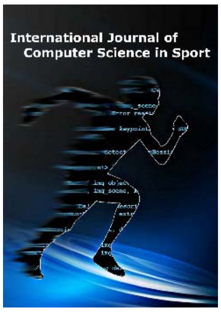
International Journal of Computer Science in Sport

Volume 15, Issue 2, 2016

Journal homepage: http://iacss.org/index.php?id=30

DOI: 10.1515/ijcss-2016-0007



Predicting Win-Loss outcomes in MLB regular season games – A comparative study using data mining methods

Soto Valero, C.

Department of Computer Science, Universidad Central “Marta Abreu” de Las Villas, Cuba

Abstract

Baseball is a statistically filled sport, and predicting the winner of a particular Major League Baseball (MLB) game is an interesting and challenging task. Up to now, there is no definitive formula for determining what factors will conduct a team to victory, but through the analysis of many years of historical records many trends could emerge. Recent studies concentrated on using and generating new statistics called sabermetrics in order to rank teams and players according to their perceived strengths and consequently applying these rankings to forecast specific games. In this paper, we employ sabermetrics statistics with the purpose of assessing the predictive capabilities of four data mining methods (classification and regression based) for predicting outcomes (win or loss) in MLB regular season games. Our model approach uses only past data when making a prediction, corresponding to ten years of publicly available data. We create a dataset with accumulative sabermetrics statistics for each MLB team during this period for which data contamination is not possible. The inherent difficulties of attempting this specific sports prediction are confirmed using two geometry or topology based measures of data complexity. Results reveal that the classification predictive scheme forecasts game outcomes better than regression scheme, and of the four data mining methods used, SVMs produce the best predictive results with a mean of nearly 60% prediction accuracy for each team. The evaluation of our model is performed using stratified 10-fold cross-validation.

KEYWORDS: MAJOR LEAGUE BASEBALL, SABERMETRICS, DATA MINING, PREDICTION, CLASSIFICATION, REGRESSION

Introduction

Sports result prediction is nowadays very popular among fans around the world, mostly due to the expansion of sports betting (Stekler, Sendor, & Verlander, 2010). Major League Baseball (MLB) is a multi-billion dollar statistically filled business, and many people are strongly interested in developing systems with the aim of providing the best prediction of the winner in many specific baseball games (Baumer & Zimbalist, 2014). Some effort has been made, but the majority of the systems created are human based benchmarked and sometimes do not work with the right dataset. Hence, users are often strongly influenced by emotions and even the experts find it difficult to select correct evaluative criteria of performance for specific teams in certain situations. One approach to surpass these and other problems in the sport forecasting domain is using data mining methods.

Data mining allows the search for valuable information in large volumes of data (Liao, Chu, & Hsiao, 2012). In particular, classification and regression methods have been widely used in predictive problems for a variety of different sport domains (Schumaker, Solieman, & Chen, 2010b). For example, Edelmann-Nusser, Hohmann, and Henneberg (2002) predict the competitive performance of an elite female swimmer (200-m backstroke) at the Olympic Games 2000 in Sydney using artificial neural networks. Morgan, Williams, and Barnes (2013) apply decision tree induction for identifying characteristics in one-versus-one player interactions that drive the outcome in hockey contests. Robertson, Back, and Bartlett (2015) use logistic regression and decision trees for explaining match outcomes in Australian Rules football. Yuan et al. (2015) present a mixture of modelers approach to forecast the 2014 NCAA men's basketball tournament.

Large quantities of historic baseball data are currently available (often publicly available) from different sources in the form of numerically or symbolically represented statistics (e.g., general season information, play-by-play, game logs, players line-up, etc.). However, despite the abundance of studies performed in the realm of financial baseball modeling, baseball games prediction have received relatively little attention in the data mining and sports informatics community (Sykora, Chung, Folland, Halkon, & Edirisinghe, 2015). Many studies have been written about the economic efficiency of baseball markets (Baumer & Zimbalist, 2014; Chang & Zenilman, 2013; Sauer, Waller, & Hakes, 2010; Witnauer, Rogers, & Saint Onge, 2007), but they lack the evaluation of the actual (or implied) outcome of games, thus studies have been emphasizing more on profitability rather than predictability.

On the other hand, there have been some attempts to measure the impact of variables associated with baseball games using numerical models, with the aim of using these factors for a wide evaluation of the team performance (Menéndez, Vázquez, & Camacho, 2015; Yang & Swartz, 2004). Nowadays, sabermetrics has been consolidated as the science of learning about baseball through objective evidence, suggesting answers to difficult questions such as: "How many home runs will some player hit next year?", "Is it easier to hit home runs in particular ballparks?" or "Are particular players especially good in clutch situations?" (Wolf, 2015). Many of the sabermetrics studies regarding to prediction are based on ranking and evaluating players individually, mainly for commercial purposes (Ockerman & Nabity, 2014; Robinson, 2014). It is a general agreement that predicting game outcomes is one of the most difficult problems on this field.

In this paper, we perform a study using four data mining methods (lazy learners, artificial neural networks, support vector machines and decision trees) and ten years of MLB regular season game data obtained from publicly available sources. Our objective consists in evaluating the predictive capabilities of these methods for both classifications (win or loss for the home team) and regression (runs difference between the home and visitor teams) schemes in the MLB games outcome prediction problem using sabermetrics statistics.

Methods

Data mining is an experimental science, and there can be no “universal” best algorithm (Wolpert & Macready, 1997). Accordingly, this study follows the CRISP-DM methodology (Shearer, 2000), which provides a structured way of conducting the data mining analysis, with the consequent improvement in the probability of obtaining accurate and reliable results. It consists in six main steps (Delen, Cogdell, & Kasap, 2012): (1) understanding the problem’s domain and defining the objectives of the study; (2) identifying, accessing and understanding the data sources; (3) preprocessing the relevant data; (4) developing the model using comparable analytical techniques; (5) evaluating and assessing the validity and utility of the model against each other and against the objectives of the study and (6) deploying the model for its use in decision-making processes.

First, we obtained data from two of the most popular and free of cost MLB data sources: the non-profit baseball organization called Retrosheet, and the Lahman Database (see Appendices). During the data preparation process, popular sabermetrics statistics were calculated and added to data. Next, feature selection algorithms are used in order to obtain a minimum set of original features, which increase the learning accuracy of the data mining algorithms and also improve the results comprehensibility (Han & Kamber, 2006).

We used four popular data mining methods and compared them with each other according to a useful predictive methodology proposed by Delen, Cogdell, and Kasap (2012). Four predictive algorithms were selected because of their capability to model both classification and regression predictive schemes, and also due to their popularity in the recent data mining literature (Liao et al., 2012).

Stratified 10-fold cross-validation methodology was employed in order to objectively assess the predictive capabilities of the different data mining algorithms and schemes. The layout of the general predictive model proposed is illustrated in Figure 1 and the parts of its structure are explained in the following subsections.

Data management

The MLB has 30 teams divided into two leagues, the American League (AL) and the National League (NL). Each team plays 162 games during the regular season (April through early October), which does not include the pre-season and the playoffs games.

In baseball, individual players are usually chosen based on popular sabermetrics statistics such as On-Base Plus Slugging (OPS), Fielding Independent Pitching (FIP) or Ultimate Zone Rating (UZR) (Soto Valero & González Castellanos, 2015). There are communities of fans who closely follow this statistics play-by-play with the objective of analyzing the contribution of individual players to their corresponding teams (Chang & Zenilman, 2013). Also, researchers have created statistics for measuring the probability for a team to win a specific game, such as the Pythagorean Expectation (PE) shown in Equation 1 (Rosenfeld, Fisher, Adler, & Morris, 2010).

$$\text{PE}=\frac{\text{runs scored}^2}{\text{runs scored}^2+\text{runs allowed}^2} \tag{1}$$

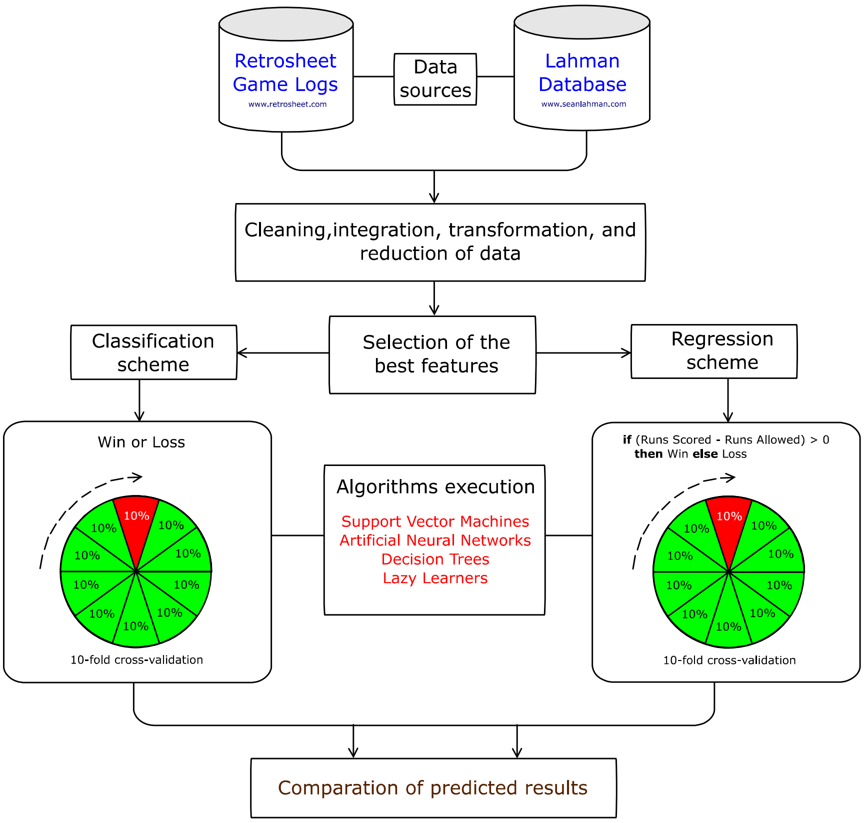


Figure 1: A graphical representation of the general predictive model applied in this study.

Exercising the data mining guidelines, we included as much relevant information in our model as we could. Data was retrieved from Retrosheet game logs for year 2005 to 2014, totaling 1620 individual game records for each team. We created a Java library in order to parse and pre-process the data for using in our data mining schemes. Each individual Retrosheet game log contains 161 data fields, but our parser removed identifying information for individual batters, coaches, and umpires. In doing this, we distilled each game down to a record of the starting pitchers, accompanied by an array of team offensive, defensive, and pitching statistics of each game. These statistics are then organized and aggregated for each season to produce a day-by-day set of accumulative statistics for each team during ten regular seasons.

During the data pre-processing stage, we reformulate the game records such that the game of each team represents a history of statistics accumulated prior to the game, rather than the statistics from the game itself. Summary of four accumulated statistics (calculated according to the previously known statistics at the moment of each game) for the 2014 MLB regular season of the San Francisco Giants and the Oakland Athletics appear in Figure 2.

In order to represent the game-related comparative characteristics of the two opponent teams in the input variables, we calculated and used the differences between the accumulative statistics of both home and visiting teams (i.e., Won Percentage, Pythagorean Expectation, On-Base Plus Slugging, Runs Created, etc.). All of these features are represented from the home team's perspective. For example, the variable WPDiff (Won Percentage Difference) represents the difference between the home team's won percentage and the visitor team's won percentage.

The output variable represents whether the home team win or lose the baseball game. That is, if the RunsDiff variable (regression scheme) takes a positive integer value, then the home team is expected to win the game by that difference; otherwise (if the RunDiff variable is a negative integer) the home team is expected to lose the game by that difference. In the case of the classification scheme, the value of the output variable is a nominal label, ‘‘Win’’ or ‘‘Loss’’, indicating the outcome of the game from the home team perspective.

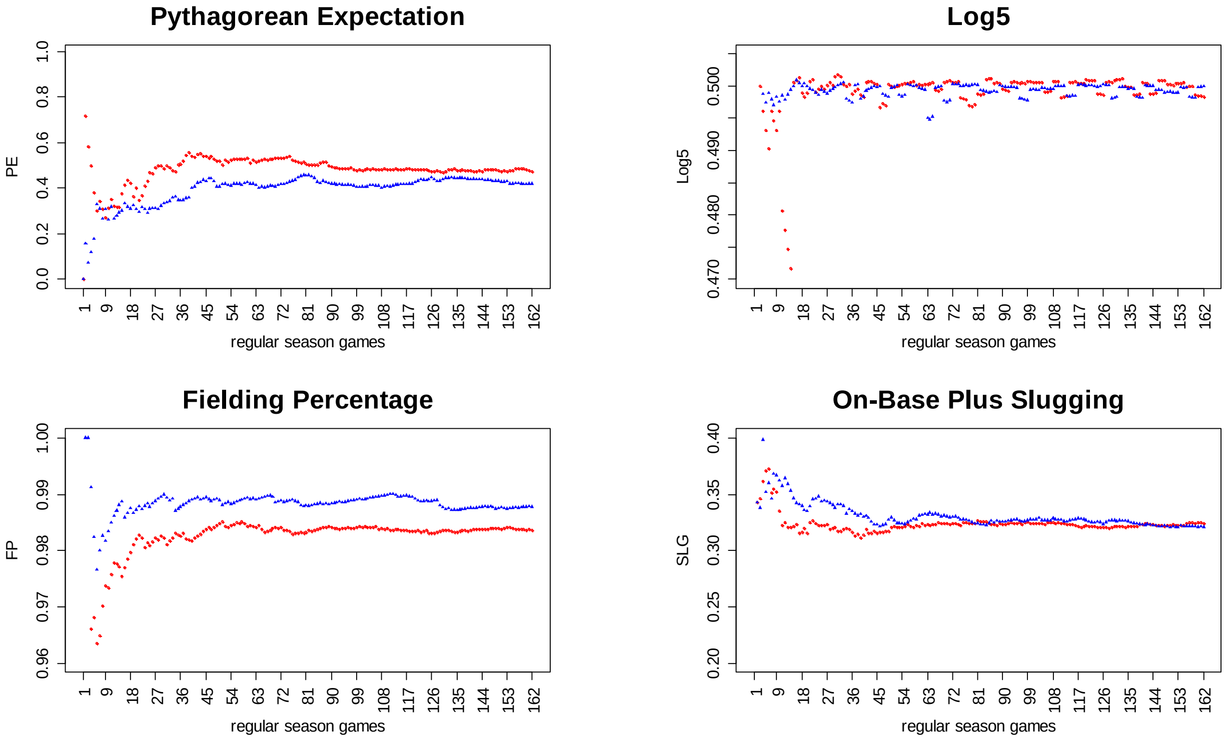


Figure 2: Example of four accumulated statistics corresponding to the San Francisco Giants (blue triangles) and Oakland Athletics (red circles) during the 2014 MLB regular season.

Data contamination

In sports forecasting, data contamination occurs when the selecting data for the predictive model is informed by “knowledge of the future” (Yuan et al., 2015). If the model’s results include statistics that were incorporated in previous data, then the input features essentially already “know” the future that they aim to predict, making the model unsuitable for true future predictions.

Contamination is a serious problem if the data on which one trains a data mining method contains implicit information from the data on which one predicts future results. For instance, running a feature selection algorithm on baseball reveals that a strong predictive statistic for outcome prediction is the length of the game (in outs). This occurs because a normal baseball game consists of 9 innings for a total of 27 outs, but games can be extended to so called extra innings. In this situation, it is proven that the home team has an important advantage. As such, the number of outs in the game is an excellent but contaminated predictor of the game outcome and should not be considered as a predictor variable.

As was stated before, attempting to use contaminated data can easily lead to disastrous results in actual forecasting. Our model avoids this problem by using only past data to train their predictive schemes. Each instance which represents a played game is built from previous statistics of the home and visitor teams, as Figure 2 shows.

Feature subset selection

Feature subset selection decreases the dataset dimension by removing irrelevant and redundant features from data. Through the acquisition of a minimum set of the original features, this technique enables data mining algorithms to operate faster and more effectively, while improving results and comprehensibility of the model (Han & Kamber, 2006).

Our generated feature set is fairly large, and it is not easy to distinguish which features are the most important for the prediction task. Also, it is possible that many of these features may be irrelevant or redundant (Trawiński, 2010). We use WEKA (Waikato Environment for Knowledge Analysis), a non-commercial and open-source data mining benchmark, for carrying out our feature selection process (M. Hall et al., 2009).

This feature selection process has three basic stages: generation, evaluation and stopping criteria (Figure 3). The validation stage, which checks the validity of the selected subset and compares the results to find the best feature subset, may not be a stage of the process (Dash & Liu, 2003).

First, the original feature set is inputting, which includes a number of features or input variables. Then the first stage of feature selection begins, which is called subset generation, where a search strategy is used for producing possible feature subsets of the original feature set for its evaluation. There are several search procedures to find the optimal subset of the original feature set (Dash & Liu, 2003). In this study, as is shown in Figure 3, the *attribute ranking* technique of WEKA is chosen for this task.

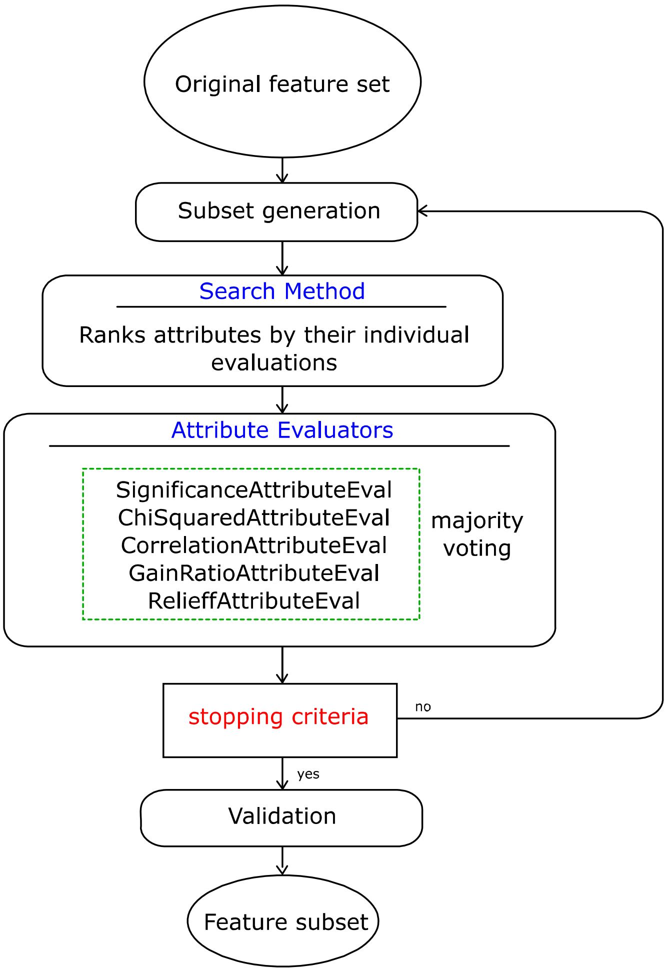


Figure 3: Feature selection process applied in this study.

Once the candidate subsets are generated, evaluation algorithms determine the best features subset and majority voting is used in order to select the most representatives. Table 1 describes the five algorithms selected for performing the attribute evaluation during the ranking process.

Table 1: Description of the attribute evaluator methods used in this study.

Attribute evaluators	Description
SignificanceAttributeEval	Evaluates the worth of an attribute by computing the probabilistic significance as a two-way function (Ahmad & Dey, 2005).
ChiSquaredAttributeEval	Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.
CorrelationAttributeEval	Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.
GainRatioAttributeEval	Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.
ReliefAttributeEval	Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class (Robnik-Šikonja & Kononenko, 1997).

A stopping criterion is needed for stopping the search and preventing an exhaustive search of subsets. The feature selection process stops by outputting the selected subset of features, which is then validated (M. A. Hall & Holmes, 2003).

After the application of this technique, our final reduced datasets contains 60 input variables. Table 2 shows the first 15 selected features after averaging the ranks obtained from the five evaluation methods using the majority vote procedure. Our feature selection model suggests that the first and most important feature for predicting outcomes in baseball are the well-known home field advantage (Smith & Groetzinger, 2010), followed by the Log5 and the Pythagorean Expectation statistics respectively.

Table 2: Ranking list of the first 15 selected features, numeric attributes represents the difference between the home and visitor team accumulative statistics.

Ranking	Feature	Type	Description
1	isHomeClub	Nominal	If the home team plays as the home club or not
2	Log5	Numeric	Log5 difference
3	PE	Numeric	Pythagorean Expectation difference
4	WP	Numeric	Won percentage for current season difference
5	RC	Numeric	Runs Created difference
6	HomeWonPrev	Nominal	If the home team won the previous game or not
7	VisitorWonPrev	Nominal	If visitor team won the previous game or not
8	BABIP	Numeric	BABIP difference
9	FP	Numeric	Fielding Percentage difference
10	PitchERA	Numeric	Starting pitchers ERA difference
11	OBP	Numeric	On-base plus slugging difference
12	Slugging	Numeric	Slugging difference
13	HomeVersusVisitor	Nominal	Particular results between home team and visitor team
14	Stolen	Numeric	Stolen bases difference
15	VisitorLeague	Nominal	The visitor team League

Datasets complexity

Data mining prediction problems are usually difficult to afford for many reasons. In certain problems the attributes are ambiguous either intrinsically or due to inadequate feature measurement and this can occur regardless of the training data size or the feature space dimensionality. Many problems have a complex decision boundary and subclass structures, thus no compact description of the class boundary is possible. Also, small sample size and dimensionality in data introduce another layer of difficulty through a lack of constraints on the generalization rules.

Often, a classification problem becomes difficult because of a mixture of these effects. Sampling density is more critical for an intrinsically complex problem than an intrinsically simple problem (e.g., a linearly separable problem with wide margins). If the sample is too sparse, an intrinsically complex problem may appear deceptively simple.

The empirically observed behavior of individual classifiers is strongly data dependent and a better understanding of such data dependency is critical for prediction. One practical measure of problem difficulty is the error rate or accuracy of a chosen classifier. However, in order to describe the real complexity of the problem and since our eventual goal is to study the behavior of various data mining methods for prediction, we need to find other measures that are independent of such choices. Previous works highlight the idea that a single descriptor may not be sufficient (Tin Kam & Basu, 2002).

Therefore, we focused on effective ways of characterizing the geometrical complexity of our predictive model. Correspondingly, we selected two geometry or topology based measures of data complexity: the maximum Fisher's discriminant ratio (F1), and the maximum feature efficiency (F3). Our study explores the distribution of MLB statistical features in the data space, in order to estimate the geometrical or topological complexity of data for the classification task (win or loss), as we believe that cluster structures can be essential characteristics for this particular problem.

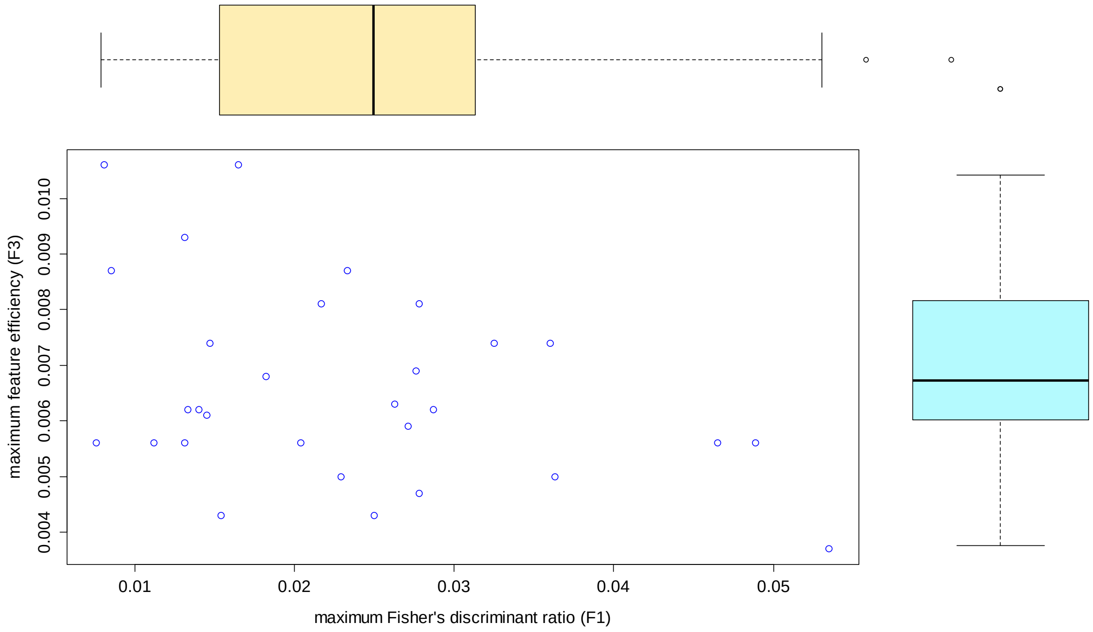


Figure 4: Scatter plot between F1and F3 values corresponding to the 30 considered datasets.

We use the KEEL Metrics-DC implementation (Alcalá-Fdez et al., 2008), in order to calculate the F1 and F2 values for each team data. Figure 4 shows the scatter plot with the results for the 30 MLB datasets of this study. It is easy to note the extreme complexity of this prediction problem, which can be seen in the low values of F1 and F3 measures (mean values of 0.02 and 0.007 respectively).

Algorithms

Lazy learning

Unlike eager learning methods, which produce a generalization as soon as the data has been loaded, lazy learning methods only approximate the function locally, and the whole computation is delayed until the execution of the classification query (Han & Kamber, 2006). Because lazy learners store the training instances, they are also referred to as instance-based learners, even though all learning is essentially based on instances.

The *k*-Nearest-Neighbors algorithm (*k*-NN) is one of the most popular instance-based learning algorithms. It is based on learning by analogy, that is, by comparing a given instance with the training instances that are similar to it. The training instances are described by its n attributes. Each instance represents a point in an n-dimensional space, and all of the training instances are stored in an n-dimensional pattern space. When an unknown instance is given, the *k*-NN classifier searches the pattern space for the *k* training instances that are closest to the unknown instance. These *k* instances are the *k* "nearest neighbors" of the unknown instance.

The "closeness" between instances is defined in terms of a distance measure, being the Euclidean Distance the standard choice. Let $I_1 = \left(I_{1_1}, I_{1_2}, \ldots, I_{1_n}\right)$ and $I_2 = (I_{2_1}, I_{2_2}, \ldots, I_{2_n})$ be two instances. Then the Euclidean Distance (ED) is calculated using the Equation 2.

$$\mathrm{ED}(I_1, I_2)=\sqrt{\sum_{i=1}^{n}\left(I_{1_i}\text{-}I_{2_i}\right)^2} \tag{2}$$

This algorithm is clearly dependent on both the user defined metric and the value of *k*, and can be used for classification or regression. The output depends on whether the expected result is a label or a numerical value (Gutierrez-Osuna, 2002).

In *k*-NN for classification, the output is a class membership. An object is classified by a majority vote over its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor. In *k*-NN for regression, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors. In the most standard case, when $k = 1$ and the distance measure is the Euclidean Distance, the *k*-NN method is known as 1-NN.

Artificial neural networks

Artificial Neural Networks (ANNs) have been widely used for sport predictions (Aslan & Inceoglu, 2007; Edelmann-Nusser et al., 2002; Young, Holland, & Weckman, 2008). The method is a product of early artificial intelligence work aimed at modeling the inner workings of the human brain, as a way of creating intelligent systems. ANNs have proven being useful because of its capacity for modeling any given function, which make it specially convenient for classification and regression tasks (Haykin, 2008).

In this study a feed-forward ANN model known as Multi Layer Perceptron (MLP) is used. This is a special ANNs algorithm that maps sets of input data onto a set of appropriate outputs using a technique called backpropagation for training the network (Han & Kamber, 2006). A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function (Figure 5). MLP has proven, given an appropriate topology, to be capable of achieving cutting-edge performance on several learning tasks (Hornik, Stinchcombe, & White, 1990).

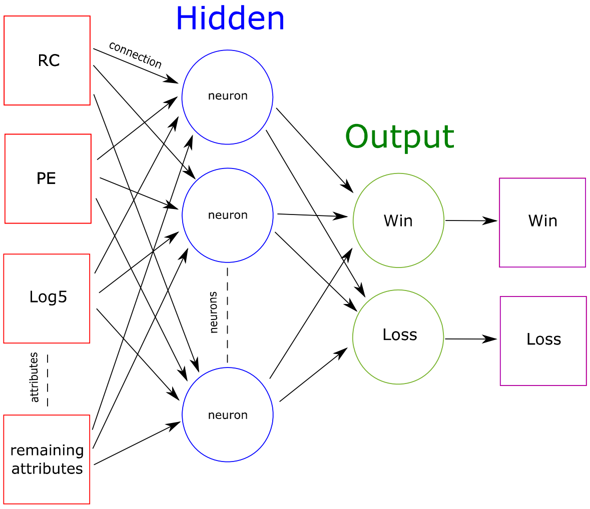


Figure 5: MLP topology with a hidden layer used in this study. The network is an interconnected group of nodes, comparable to the neurons in a brain.

Decision trees

Decision Trees (DTs) are one of the simplest and yet most successful methods used in data mining analysis (Loh, 2014). One of its main capabilities is that they can be transformed into human understandable rules, making DTs better than other data mining methods for understanding and deployment (Morgan et al., 2013; Robertson, Back, & Bartlett, 2015). A decision tree represents a function that takes as input a vector of attribute values and returns a "decision", in a form of a single output value. DTs make their decisions by performing a sequence of tests. Each internal node in the tree corresponds to a test of the value of one of the input attributes A_i and the branches from the node are labeled with the possible values of A_i. Each leaf node in the tree specifies a value to be returned by the function.

In order to construct a tree, DTs recursively separate instances into branches with the purpose of achieving the highest possible prediction accuracy. In doing so, different mathematical criteria (e.g., information gain, variance reduction, gini index, etc.) are used in order to split the set of instances into two or more subgroups (Han & Kamber, 2006). This is a recursive process, which is repeated gradually until the entire tree has been built.

In this study, we use the Weka REPTree implementation of decision trees because of its capabilities of modeling both classification and regression type prediction problems (M. Hall et al., 2009). The algorithm is a fast decision tree learner, which builds a decision/regression tree using information gain/variance and prunes it using reduced error pruning (with backfitting). This method only sorts values for numeric attributes once.

Support vector machines

Support Vector Machines (SVMs) are a powerful supervised learning method used in data analysis and pattern recognition, which also have been widely adopted for prediction in a variety of sport domains (Demens, 2015; Haghighat, Rastegari, & Nourafza, 2013; Schumaker, Solieman, & Chen, 2010a). SVMs can be used for classification and regression. The method performs a nonlinear mapping in order to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal hyperplane which separates the instances of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane.

A SVM model represents the data as points in space, mapped in a manner that the examples of the separate categories are divided by a clear gap that is as wide as possible. SVMs uses nonlinear kernel functions to transform the input data (inherently representing highly complex nonlinear relationships) to a high dimensional feature space in which the input data become more controllable (i.e., more linearly representable) than in the original input space (Burges, 1998).

While an ANN may suffer from multiple local minima, the SVMs solution is global and unique (Fischer et al., 2011). Other advantages of SVMs are that they have geometric interpretation and give a sparse solution (unlike ANNs, the computational complexity of SVMs does not directly depend on the dimension of the input space). On the other hand, they use structural error minimization, while ANNs use empirical error minimization (one of the reasons for which they are less prone to overfitting).

In this study we use the SMO algorithm, which is an improved implementation of the John Platt's sequential minimal optimization algorithm for training SVMs (Keerthi, Shevade, Bhattacharyya, & Murthy, 2001). It normalizes all attributes by default. Accordingly, the coefficients in the model are based on the normalized data, not the original data.

Evaluation

Cross-validation

The traditional method for evaluating and comparing the predictive accuracies of two or more data mining algorithms is called holdout. It consists in splitting the data into two subsets for training and testing. Often, two thirds of the instances are used for model building and the rest is used for testing.

However, the holdout method is often sampling biased, no matter what type of random sampling technique is used. In order to avoid this disadvantage, we used the 10-fold cross-validation methodology as our evaluation method (Han & Kamber, 2006). This is a popular statistical technique that is commonly used in data mining for comparing the predictive accuracies of multiple methods, which has become the standard in practical terms (Witten, Frank, & Hall, 2011).

During the 10-fold cross-validation procedure, the original set of instances is separated randomly into 10 partitions or "folds" of approximately equal size. Training and testing is performed 10 times. In iteration i, partition P_i is reserved as the test set, and the remaining partitions are collectively used to train the model. That is, in the first iteration, subsets $(P_2, \ldots, P_{10})$ collectively serve as the training set in order to obtain a first model, which is tested on P_1; the second iteration is trained on subsets $(P_1, P_3, \ldots, P_{10})$ and tested on P_1; and so on. The cross-validation estimate of the overall accuracy is calculated as the average of the 10 individual accuracy measures (Equation 3). Here, tenFoldCV is the overall accuracy of the model and F_i is the individual accuracy of each fold.

$$\text{tenFoldCV}=\frac{\sum_{i=1}^{10} F_i}{10} \quad (3)$$

The overall cross-validation accuracy relies on the random assignment of the individual instances to the different folds. Due this situation, a technique known as stratification ensures that each fold has the right proportion of each class value. Experimental studies have shown that the stratified 10-fold cross-validation procedure gives a very good estimate of the true accuracy (even if computation power allows using more folds) due to its relatively low bias and variance (Zeng & Martinez, 2000).

Performance measure

We use the accuracy measure of performance (Equation 4), in order to compare the predictive capabilities of the selected algorithms. The accuracy measures the proportion of correctly predicted games, thus forecasting the overall probability of correct classification.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \textbf{(4)}$$

Here, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) mean correct prediction of the baseball team wins, correct prediction of the baseball team losses, incorrect prediction of losses as wins, and incorrect prediction of wins as losses, respectively.

Table 3: Accuracy obtained for all algorithms and prediction schemes, highest values are highlighted in bold.

Team	1-NN		MLP		REPTree		SMO	
	Classification scheme	Regression scheme	Classification scheme	Regression scheme	Classification scheme	Regression scheme	Classification scheme	Regression scheme
ANA	56.16	56.55	**58.07**	55.62	57.18	55.18	56.57	57.16
ARI	55.34	55.49	56.22	53.63	58.67	56.67	**59.12**	58.53
ATL	56.22	56.05	57.19	53.01	56.96	**58.29**	57.43	56.17
BAL	55.55	54.53	56.33	56.58	57.79	58.51	**59.25**	54.21
BOS	56.77	57.04	58.82	58.72	57.58	58.60	**59.58**	56.91
CHA	57.12	57.07	56.76	56.64	56.81	54.78	57.88	**58.00**
CHN	54.97	54.68	56.10	54.25	56.65	55.12	**57.04**	56.42
CIN	56.78	56.83	58.41	58.63	59.72	59.68	**60.13**	59.61
CLE	56.30	55.86	58.37	56.92	58.07	59.03	**60.75**	58.96
COL	58.92	58.38	61.41	55.90	58.76	**62.29**	61.50	59.99
DET	55.64	56.52	57.35	57.82	58.59	58.32	**58.64**	57.13
FLO	52.35	52.11	55.78	53.16	57.01	52.85	**56.41**	56.14
HOU	57.59	59.47	60.59	56.86	59.92	60.27	61.06	**61.82**
KCA	53.48	54.06	59.35	59.34	58.75	55.74	60.08	**60.27**
LAN	55.53	56.27	**57.46**	56.64	55.75	55.34	55.65	54.03
MIL	52.81	52.76	**58.82**	57.48	58.48	58.54	58.65	56.85
MIN	54.96	55.06	58.05	56.11	**60.78**	58.78	60.43	58.65
NYA	55.63	55.74	**60.35**	57.85	59.63	60.27	60.26	58.72
NYN	56.38	56.24	56.57	58.04	58.57	56.73	**59.65**	58.22
OAK	54.07	53.04	57.61	56.02	56.37	58.57	**58.88**	58.31
PHI	55.48	55.18	56.83	55.43	57.23	57.54	**59.18**	58.28
PIT	57.68	57.01	58.74	58.82	60.29	60.56	**62.27**	59.99
SDN	56.09	56.64	55.57	55.40	57.68	57.57	**59.21**	57.13
SEA	58.13	58.41	**58.99**	54.62	57.04	57.17	57.87	54.77
SFN	54.08	54.28	56.49	53.78	55.94	54.78	56.47	**57.72**
SLN	54.37	53.53	58.24	57.33	59.10	59.25	59.03	**59.50**
TBA	55.20	55.34	58.00	56.33	55.95	**60.05**	57.45	56.38
TEX	57.25	56.39	58.28	54.96	57.04	**58.94**	58.36	55.52
TOR	57.43	56.73	58.30	53.94	56.30	**58.91**	57.83	57.10
WAS	61.26	**62.08**	57.72	55.49	57.24	58.73	61.16	57.35
Mean	55.98	55.97	57.89	56.17	57.86	57.90	**58.92**	57.66

Results

We used the Weka Experiment Environment Interface (M. Hall et al., 2009) to facilitate the execution and experimental comparisons of performance of our predictive model. Table 3 shows the 10-fold cross-validation results, for both classification and regression based schemes, of the four data mining algorithms included in this study.

Among the four data mining methods, SVMs used both for classification and regression offered the higher prediction accuracies (Figure 6). Overall, SVMs produced a mean accuracy of 59% and 58% for classification and regression schemes respectively (STD of 1.64 and 1.82), followed by ANNs for classification with a mean accuracy of 58%, and DTs for regression with a mean accuracy of nearly 58%. It is noticeable a peak of just over 61% and 62% of prediction accuracy for the team of Washington Nationals using lazy learning (classification and regression based schemes respectively) but this can be considered as an outlier and therefore it is not representative of the whole results.

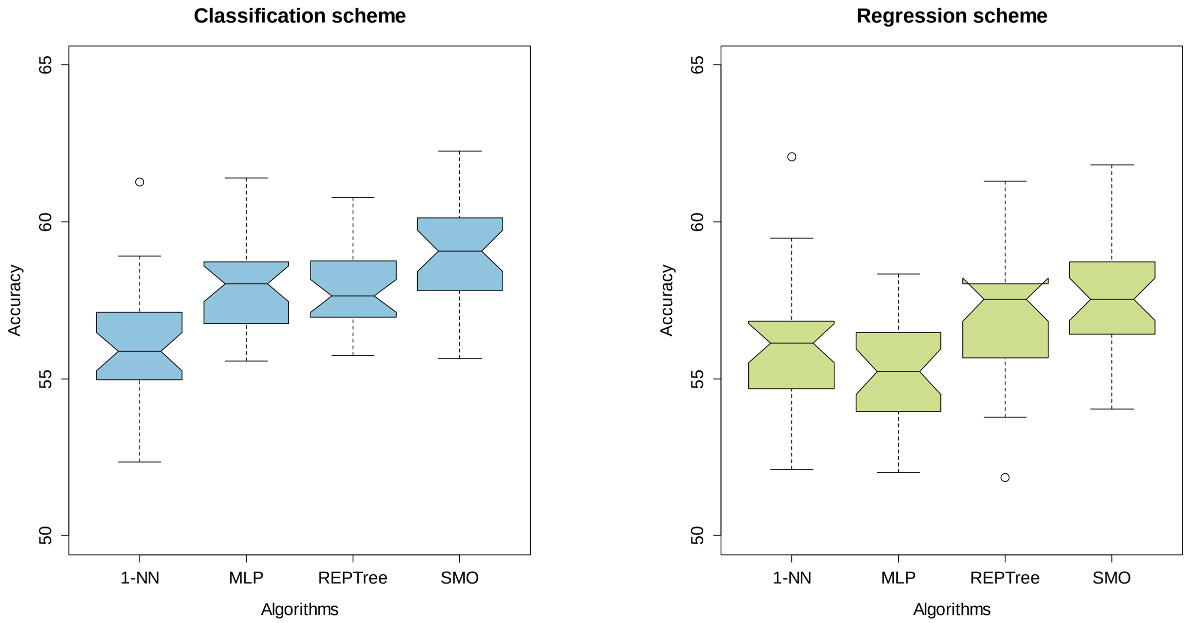


Figure 6: Box plots of accuracy obtained for both predictive schemes using the four chosen data mining methods.

We performed a comparison of the accuracy values among all the data mining methods in order to determine the most competitive ones. The Aligned Friedman test applied detects significant differences on a significant level of α = 0.05 for both schemes (for classification scheme the ρ-value is $3.06*10^{-5}$ and for regression scheme it is $2.18*10^{-5}$). Table 4 shows the ranking obtained. The most accurate algorithm is chosen as the control method for the application of the post-hoc procedure on the predictive schemes. In both cases, the model selected is SMO, while 1-NN shows the lowest value of accuracy. The Hochberg post-hoc procedure detects significant differences in favor of SMO with respect to 1-NN, MLP and REPTree, with the exception of REPTree for regression.

In order to determine the best predictive scheme, we applied a Wilcoxon Signed Ranks Test of accuracy to the SMO classification and regression outputs (the best method). The result shows that classification schemes outperform regression (Table 5). The difference obtained results significative at a significance level of α = 0.05.

Table 4: Aligned Friedman ranking of the accuracy for both predictive schemes. Adjusted p-values for the post-hoc procedure of Hochberg.

Schemes	Algorithms	Average Rank	$\rho_{Hochberg}$	Hypothesis
Classification	1-NN	97.40	0	Rejected
	MLP	57.13	0.001638	Rejected
	REPTree	58.62	0.001638	Rejected
	SMO	28.85	-	-
Regression	1-NN	70.63	0	Rejected
	MLP	86.40	0.000134	Rejected
	REPTree	50.13	0.88473	Accepted
	SMO	34.83	-	-

Table 5: Wilcoxon Signed Ranks Test of accuracy for SMO regression and classification predictive schemes.

SMO schemes	Negative Ranks	Positive Ranks	ρ_{value}
classification – regression	6	24	$7.2*10^{-5}$

As an additional evaluation of this model, we compared its predictions with regards to the betting market. For this aim, we used information from the Covers[1] odd publisher. We collected data from Moneyline odds and calculated its predictive accuracy for all MLB teams and games during the 2014 regular season. The SMO (our best method) was trained using 9 years of data (2005-2013) and tested using the 2014 season. Table 6 shows the results obtained. The Wilcoxon Signed Rank Test shows no difference between Covers predictions and the SMO classification algorithm (ρ_{value}= 0.066) at a significance level of α = 0.05. According to this result, our approach could be considered competitive with regards to the betting market.

Table 6: Comparison of classification accuracies between the Moneyline betting market and the SMO classification method for the 2014 MLB regular season.

Team	Moneyline	SMO	Team	Moneyline	SMO
ANA	60.00	56.43	MIL	52.47	53.49
ARI	58.64	55.49	MIN	59.88	55.38
ATL	52.47	51.52	NYA	55.56	59.34
BAL	50.89	54.49	NYN	59.26	60.00
BOS	50.00	54.00	OAK	63.19	49.28
CHA	56.79	52.06	PHI	51.85	58.46
CHN	56.79	53.55	PIT	61.35	56.00
CIN	57.41	64.00	SDN	54.94	60.00
CLE	56.17	56.88	SEA	53.70	47.00
COL	64.20	58.46	SFN	60.89	57.38
DET	54.55	53.04	SLN	57.31	56.43
FLO	53.09	47.00	TBA	51.23	52.08
HOU	54.94	57.47	TEX	59.26	57.88
KCA	58.19	47.40	TOR	50.62	50.22
LAN	60.24	50.67	WAS	62.65	55.69

[1] http://covers.com

Discussion

We presented a predictive model, based on the CRISP-DM methodology, in order to forecast baseball games using popular data mining methods. Our model employs stratified cross-validation as evaluation criteria, and has the advantage that it could be easily expanded with more data mining methods and replicated in other sports when sufficient data is available.

We tested our model by using ten years of MLB records and predicting outcomes for each MLB team separately. Data contamination was avoided by using predictors based only in past data. We surpassed some incompatibilities in the Retrosheet and Lahman databases, generating the largest possible set of sabermetrics statistics from raw data.

Feature subset selection, based on a majority vote ranking procedure with five attribute evaluation methods, was applied for improving prediction results. Home club field advantage, Log5 and the Pythagorean Expectation (in this order) were selected as the most important features for our prediction task (Table 2). Experimental empirical studies performed suggest that the addition of more features do not improve the accuracy of our model.

This study demonstrates the inherent complexity in predicting outcomes in MLB regular season games (Figure 4). The four popular data mining methods applied show accuracy values just under 60% (Figure 6), which represents an improvement over random guessing but it is not really a remarkable result in betting context (even when the most novel sabermetrics statistics were used as base predictors).

The implementation of other data mining methods (rule based, rough sets, genetic algorithms, ensemble models, etc.) and the addition of more features from other baseball data sources such as Baseball Reference[2] and PITCHf/x[3] may produce somewhat different results. But due to the complexity and extension of the analyzed datasets, we strongly believe that more accurate results, when only statistical data is used, may not be feasible for predicting outcomes in MLB. In this sense, experimentation with other amateur and professional baseball leagues, such as the Korea Professional Baseball or the Nippon Professional Baseball, are needed in order to generalize this criterion.

In the author’s opinion, modeling baseball games as a stochastic process and applying dynamic learning using “within-game” data should bring better predictive results than general models (Percy, 2015). In this sense, data mining methods have achieved success for selecting strategies and predicting outcomes in the context of some specific baseball game situations. For example, assessing pitcher and catcher influences on base stealing (Loughin & Bargen, 2008); determining when a starting pitcher should be relieved (Gartheeban & Guttag, 2013); and predicting the probability of a strikeout for a particular batter/pitcher matchup (Healey, 2015). However, to create general and accurate models for predicting MLB game outcomes is still an open field of research in the sports analytics domain today.

In the future, we will focus on different improvements in our model deployment: adjusting parameters, refining features and extending datasets. Also, we plan to evaluate the proposed predictive model in teams of other sports such as basketball, football and water polo. We believe that this model can be useful for teams in certain seasonal phases and against specifics opponent teams. This is because we suppose that it is in specific baseball game environments that coaches could most effectively take advantage of our model, in order to translate statistical knowledge into team wins.

[2] http:// baseball-reference.com

[3] http://gd2.mlb.com/components/game/mlb

Conclusion

This paper compares the performance of four different data mining methods in the context of predicting outcomes (win or loss) for independent MLB regular season games. First, we proved the inherent difficulty of this particular prediction problem by showing and characterizing its complexity. In order to test our predictive model, we used sabermetrics statistics to measure teams performance and created a total of 30 datasets (one for each MLB team), corresponding to ten years of free available data (between 2005 and 2014 inclusively). Feature selection methods applied show that the most important predictor variable is the home field advantage. Four popular data mining methods were applied to reduced datasets (classification and regression based) and where evaluated using the 10-fold cross-validation criterion. Overall, classification schemes outperform the regression based schemes and SVMs results the best predictor method, with accuracy values of nearly 60%. In spite of results that were not surprisingly accurate, they became a good starting point for future works in this field. The application of this predictive model to an extended set of features and data could provide not only a basis for prediction, but also revealed potential strengths and weaknesses of individual teams by quantifying their win perspectives. The model could also show the significance of specific statistics and its relevance to victory, which is invaluable to MLB managers and enthusiasts of this sport.

References

Ahmad, A., & Dey, L. (2005). A feature selection technique for classificatory analysis. *Pattern Recognition Letters, 26*(1), 43-56. doi: 10.1016/j.patrec.2004.08.015

Alcalá-Fdez, J., Sánchez, L., García, S., Jesus, M. J., Ventura, S., Garrell, J. M., . . . Herrera, F. (2008). KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing, 13*(3), 307-318. doi: 10.1007/s00500-008-0323-y

Aslan, B. G., & Inceoglu, M. M. (2007). *A comparative study on neural network based soccer result prediction.* Paper presented at the Seventh International Conference on Intelligent Systems Design and Applications.

Baumer, B., & Zimbalist, A. (2014). Quantifying Market Inefficiencies in the Baseball Players' Market. *Eastern Economic Journal, 40*(4), 488-498. doi: 10.1057/eej.2013.43

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery, 2*(2), 121-167. doi: 10.1023/a:1009715923555

Chang, J., & Zenilman, J. (2013). A study of sabermetrics in Major League Baseball: The impact of Moneyball on free agent salaries.

Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence, 151*(1–2), 155-176. doi: 10.1016/S0004-3702(03)00079-1

Delen, D., Cogdell, D., & Kasap, N. (2012). A comparative analysis of data mining methods in predicting NCAA bowl outcomes. *International Journal of Forecasting, 28*(2), 543-552. doi: 10.1016/j.ijforecast.2011.05.002

Demens, S. (2015). Riding a probabilistic support vector machine to the Stanley Cup. *Journal of Quantitative Analysis in Sports, 11*(4), 205-218. doi: 10.1515/jqas-2014-0093

Edelmann-Nusser, J., Hohmann, A., & Henneberg, B. (2002). Modeling and prediction of competitive performance in swimming upon neural networks. *European Journal of Sport Science, 2*(2), 1-10. doi: 10.1080/17461390200072201

Fischer, A., Do, M., Stein, T., Asfour, T., Dillmann, R., & Schwameder, H. (2011). Recognition of Individual Kinematic Patterns during Walking and Running-A

Comparison of Artificial Neural Networks and Support Vector Machines. *International Journal of Computer Science in Sport, 10*(1).

Gartheeban, G., & Guttag, J. (2013). *A data-driven method for in-game decision making in MLB: when to pull a starting pitcher.* Paper presented at the Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.

Gutierrez-Osuna, R. (2002). The k nearest neighbor rule (k-nnr). *k-NN Lecture Notes*.

Haghighat, M., Rastegari, H., & Nourafza, N. (2013). A review of data mining techniques for result prediction in sports. *Advances in Computer Science: an International Journal, 2*(5), 7-12.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl., 11*(1), 10-18. doi: 10.1145/1656274.1656278

Hall, M. A., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on, 15*(6), 1437-1447. doi: 10.1109/TKDE.2003.1245283

Han, J., & Kamber, M. (2006). *Data Mining Concepts and Techniques* (2nd ed.): Morgan Kaufmann Publishers.

Haykin, S. (2008). *Neural networks and learning machines* (3rd ed.). New Jersey: Prentice Hall.

Healey, G. (2015). Modeling the Probability of a Strikeout for a Batter/Pitcher Matchup. *Knowledge and Data Engineering, IEEE Transactions on, 27*(9), 2415-2423. doi: 10.1109/TKDE.2015.2416735

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks, 3*(5), 551-560. doi: 10.1016/0893-6080(90)90005-6

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation, 13*(3), 637-649. doi: 10.1162/089976601300014493

Liao, S.-H., Chu, P.-H., & Hsiao, P.-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications, 39*(12), 11303-11311. doi: 10.1016/j.eswa.2012.02.063

Loh, W.-Y. (2014). Fifty Years of Classification and Regression Trees. *International Statistical Review, 82*(3), 329-348. doi: 10.1111/insr.12016

Loughin, T. M., & Bargen, J. L. (2008). Assessing pitcher and catcher influences on base stealing in Major League Baseball. *Journal of sports sciences, 26*(1), 15-20. doi: 10.1080/02640410701287255

Menéndez, H. D., Vázquez, M., & Camacho, D. (2015). Mixed Clustering Methods to Forecast Baseball Trends. In D. Camacho, L. Braubach, S. Venticinque & C. Badica (Eds.), *Intelligent Distributed Computing VIII* (pp. 175-184). Cham: Springer International Publishing.

Morgan, S., Williams, M. D., & Barnes, C. (2013). Applying decision tree induction for identification of important attributes in one-versus-one player interactions: A hockey exemplar. *Journal of sports sciences, 31*(10), 1031-1037. doi: 10.1080/02640414.2013.770906

Ockerman, S., & Nabity, M. (2014). Predicting the Cy Young Award Winner. *PURE Insights, 3*(1), 9.

Percy, D. F. (2015). Strategy selection and outcome prediction in sport using dynamic learning for stochastic processes. *Journal of the Operational Research Society, 66*(11), 1840-1849. doi: 10.1057/jors.2014.137

Robertson, S., Back, N., & Bartlett, J. D. (2015). Explaining match outcome in elite Australian Rules football using team performance indicators. *Journal of sports sciences*, 1-8. doi: 10.1080/02640414.2015.1066026

Robinson, S. J. (2014). Extracting Individual Offensive Production from Baseball Run Distributions. *International Journal of Computer Science in Sport, 13*(2).

Robnik-Šikonja, M., & Kononenko, I. (1997). *An adaptation of Relief for attribute estimation in regression.* Paper presented at the Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97).

Rosenfeld, J. W., Fisher, J. I., Adler, D., & Morris, C. (2010). Predicting overtime with the Pythagorean formula. *Journal of Quantitative Analysis in Sports, 6*(2). doi: 10.2202/1559-0410.1244

Sauer, R. D., Waller, J. K., & Hakes, J. K. (2010). The progress of the betting in a baseball game. *Public Choice, 142*(3-4), 297-313. doi: 10.1007/s11127-009-9544-6

Schumaker, R. P., Solieman, O. K., & Chen, H. (2010a). Greyhound racing using support vector machines. *Sports Data Mining* (pp. 117-125): Springer US.

Schumaker, R. P., Solieman, O. K., & Chen, H. (2010b). *Sports Data Mining*: Springer US.

Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. *Journal of Data Warehousing, 5*, 13–22.

Smith, E. E., & Groetzinger, J. D. (2010). Do fans matter? The effect of attendance on the outcomes of Major League Baseball games. *Journal of Quantitative Analysis in Sports, 6*(1). doi: 10.2202/1559-0410.1192

Soto Valero, C., & González Castellanos, M. (2015). Sabermetría y nuevas tendencias en el análisis estadístico del juego de béisbol [Sabermetrics and new trends in statistical analysis of baseball]. *Retos, 28*(2), 122-127.

Stekler, H. O., Sendor, D., & Verlander, R. (2010). Issues in sports forecasting. *International Journal of Forecasting, 26*(3), 606-621. doi: 10.1016/j.ijforecast.2010.01.003

Sykora, M., Chung, P. W. H., Folland, J. P., Halkon, B. J., & Edirisinghe, E. A. (2015). Advances in Sports Informatics Research *Computational Intelligence in Information Systems* (pp. 265-274): Springer.

Tin Kam, H., & Basu, M. (2002). Complexity measures of supervised classification problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(3), 289-300. doi: 10.1109/34.990132

Trawiński, K. (2010). *A fuzzy classification system for prediction of the results of the basketball games.* Paper presented at the Fuzzy Systems (FUZZ), 2010 IEEE International Conference.

Witnauer, W. D., Rogers, R. G., & Saint Onge, J. M. (2007). Major league baseball career length in the 20th century. *Population research and policy review, 26*(4), 371-386. doi: 10.1007/s11113-007-9038-5

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques* (3rd ed.): Morgan Kaufmann Publishers.

Wolf, G. H. (2015). The Sabermetric Revolution: Assessing the Growth of Analytics in Baseball by Benjamin Baumer and Andrew Zimbalist (review). *Journal of Sport History, 42*(2), 239-241.

Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82. doi: 10.1109/4235.585893

Yang, T. Y., & Swartz, T. (2004). A Two-Stage Bayesian Model for Predicting Winners in Major League Baseball. *Journal of Data Science, 2*, 61-73.

Young, W. A., Holland, W. S., & Weckman, G. R. (2008). Determining hall of fame status for major league baseball using an artificial neural network. *Journal of Quantitative Analysis in Sports, 4*(4). doi: 10.2202/1559-0410.1131

Yuan, L.-H., Liu, A., Yeh, A., Kaufman, A., Reece, A., Bull, P., . . . Bornn, L. (2015). A mixture-of-modelers approach to forecasting NCAA tournament outcomes. *Journal of Quantitative Analysis in Sports, 11*(1), 13-27. doi: 10.1515/jqas-2014-0056

Zeng, X., & Martinez, T. R. (2000). Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence, 12*(1), 1-12. doi: 10.1080/095281300146272

Appendix

This appendix contains detailed information about the two freely available data sources used in this study: Retrosheet game logs and Lahman database. Baseball records from these sources have a growing level of detail, from seasonal stats available since the 1871 season, to box score data for individual games, to play-by-play accounts covering most games since 1945. It is important to analyze carefully the structure of these datasets in order to correctly understand the information they provide.

Retrosheet game logs

The Retrosheet organization was founded in 1989 with the purpose of collecting play-by-play information about every game played in the MLB history. The Retrosheet website[4] provides individual game logs data going back to 1871. A game log has details regarding when the game was played, how many spectators attended, the teams and the ballpark, and the score (both the final score and the inning by inning runs scored). In addition, the game log file includes teams offensive and defensive statistics, starting players, managers, and umpire crews. Table 7 shows details of all 161 fields compiled for each game. There are missing observations for some game log variables for earlier baseball seasons. The information used here was obtained free of charge from and is copyrighted by Retrosheet. Interested parties may contact Retrosheet at www.retrosheet.org.

Lahman database

Sean Lahman, who is an active baseball journalist and book author, makes freely available at his website[5] one of the most complete databases of baseball statistics. Lahman database provides seasonal pitching, hitting, and fielding statistics for all players in MLB from the first professional league in 1871, to the formation of MLB in 1901, to the present day. In addition, this database includes a number of supplemental tables including All-Star game appearances, Hall of Fame voting data, managerial statistics, and batting and pitching statistics for players in the post-season. The data is available in several formats: as SQL database, a set of comma-separated-value (csv) tables, and recently also as R package. Table 8 shows a description of each table in the comma-separated-value version.

[4] http://retrosheet.org/gamelogs/index.html

[5] http://seanlahaman.com

Table 7: Summary of Retrosheet game logs data fields.

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.

Table 8: Descriptions of tables in the Lahman database.

Table	**Description**
AllStarFull	Players' appearances in All-Star games.
Appearances	Seasonal players' appearances by position.
AwardsManagers	Recipients of the Manager of the Year award.
AwardsPlayers	Players' recipients of the various awards.
AwardsShareManagers	Voting results for the Manager of the Year award.
AwardsSharePlayers	Voting results for the various awards for players.
Batting	Seasonal batting statistics.
BattingPost	Seasonal batting statistics for post-season.
Fielding	Seasonal fielding statistics.
FieldingOF	Seasonal appearances at the three outfield positions.
FieldingPost	Seasonal fielding data for post-season.
HallOfFame	Voting results for the Hall of Fame.
Managers	Seasonal data for managers.
ManagersHalf	Seasonal split data for managers.
Master	Biographical information.
Pitching	Seasonal pitching statistics.
PitchingPost	Seasonal pitching statistics for post-season.
Salaries	Seasonal salaries for players.
Schools	List of college teams.
SchoolsPlayers	Information on schools attended by players.
SeriesPost	Outcomes of post-season series.
Teams	Seasonal stats for teams.
TeamsFranchises	Timelines of franchises.
TeamsHalf	Seasonal split stats for teams.