Computer Science and Software Engineering

Computer Science and Software Engineering

Technical Reports

Miami University *Year 2018*

A Survey of Baseball Machine Learning: A Technical Report

Kaan Koseler Matthew Stephan

MIAMI UNIVERSITY

OXFORD, OH • EST. 1809

DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

TECHNICAL REPORT: MU-CEC-CSE-2018-001

A Survey of Baseball Machine Learning: A Technical Report

Kaan Koseler and Matthew Stephan

This is an Author's Original Manuscript (AOM) of an article published Taylor & Francis in Applied Artificial Intelligence on February 26 2018, available online: https://doi.org/10.1080/08839514.2018.1442991

A Survey of Baseball Machine Learning: A Technical Report

Kaan Koseler and Matthew Stephan

November 22, 2017

Abstract

Statistical analysis of baseball has long been popular, albeit only in limited capacity until relatively recently. The recent proliferation of computers has added tremendous power and opportunity to this field. Even an amateur baseball fan can perform types of analyses that were unimaginable decades ago. In particular, analysts can easily apply machine learning algorithms to large baseball data sets to derive meaningful and novel insights into player and team performance. These algorithms fall mostly under three problem class umbrellas: Regression, Binary Classification, and multiclass classification. Professional teams have made extensive use of these algorithms, funding analytics departments within their own organizations and creating a multi-million dollar thriving industry. In the interest of stimulating new research and for the purpose of serving as a go-to resource for academic and industrial analysts, we have performed a systematic literature review of machine learning algorithms and approaches that have been applied to baseball analytics. We also provide our insights on possible future applications. We categorize all the approaches we encountered during our survey, and summarize our findings in two tables. We find two algorithms dominated the literature, 1) Support Vector Machines for classification problems and 2) Bayesian Inference for both classification and Regression problems. These algorithms are often implemented manually, but can also be easily utilized by employing existing software, such as WEKA or the Scikit-learn Python library. We speculate that the current popularity of neural networks in general machine learning literature will soon carry over into baseball analytics, although we found relatively fewer existing articles utilizing this approach when compiling this report.

1 Introduction

The field of baseball analytics has experienced tremendous growth in the past two decades. Often referred to as "sabermetrics", a term popularized by Bill James, it has become a critical part of professional baseball leagues worldwide [1, 2]. All teams in Major League Baseball (MLB) have established their own departments dedicated to such analysis and millions of dollars invested [3]. Popular websites such as Fangraphs [1] and Baseballsavant[2] exemplify the pop-

[1] http://www.fangraphs.com/
[2] https://baseballsavant.mlb.com/

ularity of baseball analytics among baseball fans. There is also a growing body of academic literature investigating baseball analytics.

While analyzing baseball data is nothing new, analytics incorporating machine learning (ML) techniques are emerging. Machine learning is particularly suited to the data-heavy, discrete nature of baseball [4]. Professional baseball teams now collect data on nearly every aspect of the game. For example, the PITCHf/x system[3] generates large amounts of data tracking pitched balls that are particularly useful for academic and professional analysis. Machine learning allows teams and other stakeholders to glean insights that are otherwise not readily apparent from human analysis.

In this technical report we perform a systematic literature review to categorize and summarize the applications of machine learning to baseball. Our goal is to establish the state of the art, help practitioners discover existing techniques, and guide future research. We categorize the approaches into the three problem classes defined by Smola & Vishwanathan [5]: Binary Classification, multiclass Classification, and Regression. In addition to reporting the applications, we include representative examples for each category and speculate on potential future applications. While we perform an exhaustive survey of the publicly available literature, an important caveat to consider is that the public literature is inherently incomplete. Baseball analytics is a multi-million dollar and competitive industry. Professional organizations have a strong interest in keeping their work proprietary.

We begin in Section 2 with background information on machine learning and baseball analytics to help establish the context of this report. In Section 3, we outline the protocol we use for our systematic literature review based on established guidelines. Section 4 is organized by the three problem classes. It presents our findings on existing work, examples, and speculative potential applications. We summarize our results in Section 5 and conclude in Section 6.

2 Background

This section provides the background necessary to assist readers in better understanding this report. We summarize machine learning as a whole, and overview the three problem classes we use to categorize applications. We include a brief primer on baseball analytics.

2.1 Machine Learning

The concept of machine learning has a variety of definitions. There is broad agreement that it involves automated pattern extraction from data [6]. Much of the time, the patterns extracted from machine learning techniques are used to create a model for making predictions. Most of the time, this is done through what is referred to as supervised learning. We present a high-level description in Figure 1, wherein training data is fed into an algorithm that builds a predictive model that can then be queried for predictions on new data. There are other types of learning, including unsupervised learning and reinforcement learning. However, for most practical applications, supervised machine learning is pre-

[3] http://www.sportvision.com/baseball/pitchfx

ferred and tends to yield good results [6]. We now summarize these types of learning and include simple examples.

Bishop defines supervised machine learning as problems that take in training data in the form of example input vectors, x_i, and their corresponding target vectors, y_i [7]. For example, consider the case of predicting whether a certain student will gain admittance into Miami University. A natural place to begin is an examination of past admission cycles. We might take in input vectors of student attributes like students' GPA, SAT scores, and admission status from the year 2010. The crucial marker of a supervised learning problem is the inclusion of past observations and their target vectors. Since we have our target vectors, and know what classifications we are looking for, this type of learning is considered "supervised". We provide some examples of supervised learning in our summary of machine learning classes.

Unsupervised learning can be defined through problems that take in training data input vectors, x_i, but have no corresponding target vectors [7]. Because of this, unsupervised learning is generally ignored for prediction purposes. Rather, unsupervised learning is generally used for clustering, which involves grouping similar data points together. For example, consider the performance of all basketball players in the National Basketball Association over the course of one season. We may measure their performance by calculating their points-per-game (PPG), and then cluster them into groups such as "Elite", "Good", "Average", and "Poor" classes. This is an example of an unsupervised learning problem as it includes no prediction. We are simply seeking to classify players based on the PPG metric. The algorithm we employ might classify Elite players as those with $PPG \geq 20$.

Reinforcement learning is "concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward" [7]. Reinforcement learning is often employed with game-playing Artificial Intelligence Systems. For example, Tesauro was able to demonstrate the use of reinforcement learning to create a master-level backgammon playing program [8]. Reinforcement learning generally makes use of the trial and error process to determine optimal actions. The optimal action is specific to each respective domain of interest. If, for instance, we are interested in designing a chess-playing AI, the optimal action is the one bringing the AI into a more favorable position relative to its opponent.

The data that feed programs employing machine learning can take many forms, and these different forms require different approaches. In particular, different data types will provide different types of insights depending on the requirements of the problem space. We now cover several broad classes of machine learning, with the acknowledgement that this is not an exhaustive list.

2.1.1 Binary Classification

Binary classification is perhaps the best-understood problem in machine learning [5].It is also relatively simple to explain. Given a set of observations in a domain X, determine the value, Y, for each observation, where Y is a binary value that classifies the observation. In general, the values of Y are referred to as either positive or negative. This can be modified to suit the needs of the user.

Consider, a student who submits an application to a university will either

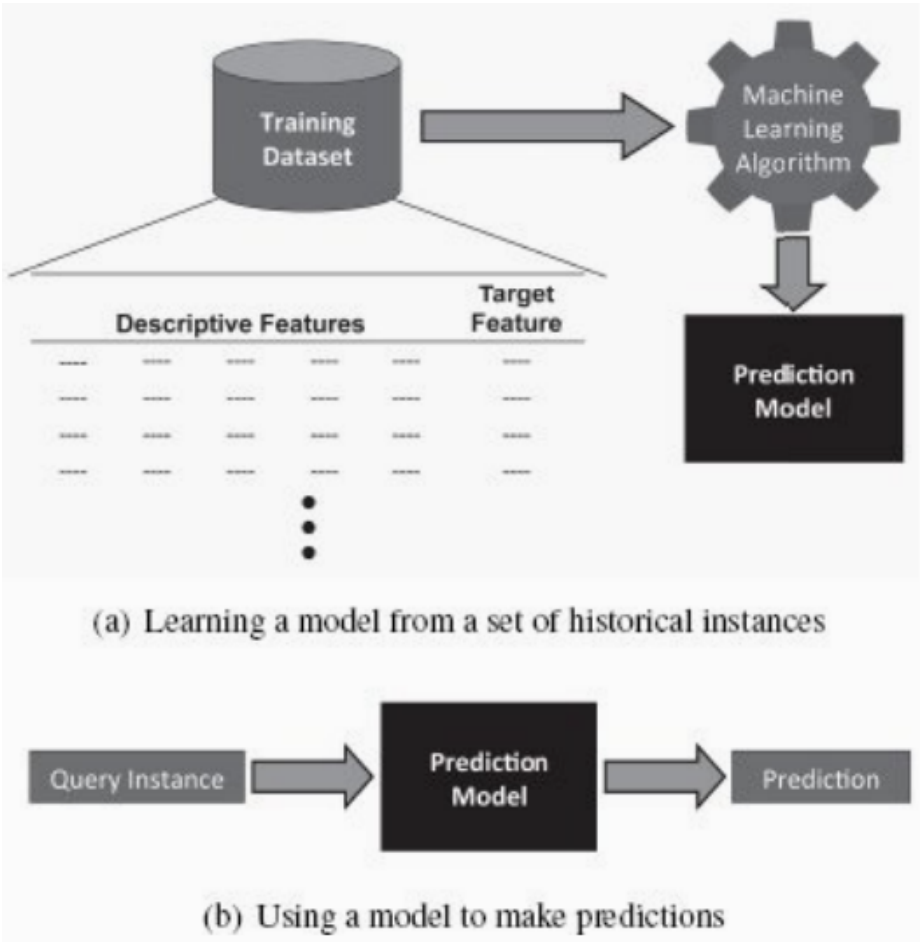

Figure 1: High-level diagram of Machine Learning [6]

be admitted or rejected. Although there are other admission categories, such as "wait list", assume for the purpose of this example that admission or rejection are the only classifiers. Admission decisions are not made arbitrarily by an institution. In our example, this university bases their decision on observational data, such as GPA or involvement with extracurricular activities. Note that these are different types of data. GPA is quantitative and extracurricular involvement being more qualitative. For example, a student who was Class President may be favored over a student who was a member of several clubs but did not hold any leadership positions. Such data are, by definition, difficult to quantify. The more important idea is that there are usually several data points for each observation. In this example, while a high GPA may increase a student's probability of admission, rarely is it the sole factor in the decision.

There are several different algorithms used to classify data into two groups. One of the more commonly used is the Support Vector Machine (SVM) algorithm. A SVM searches data for a decision boundary that can best separate the data into two classes [6]. This decision boundary is defined by the margins

Table 1: Simple example of a classification problem

GPA	SAT Score	Admission Status
3.8	1850	Admit
2.9	2030	Reject
2.75	2180	Admit
3.33	1960	Reject
3.5	1710	? (Admit/Reject)

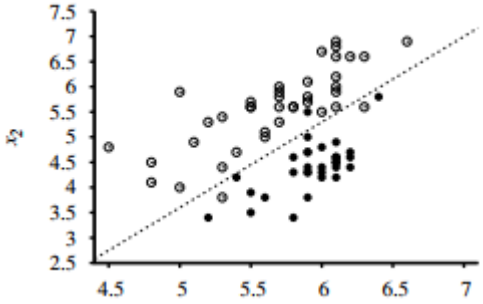

Figure 2: Example of a basic Linear Classifier [9]

around it, which themselves are defined by data points referred to as support vectors. The example in Figure 2 can be thought of as having a decision boundary with no margins. We may also make use of the Perceptron Learning Rule, which uses a set of weights that are continually updated as we iterate through the data [9]. These algorithms may not always lead to a perfect classifier if the data are not linearly separable, which indicates significant overlap between the classes.

We may also use decision trees for Binary Classification [9]. A decision tree works by taking in a vector of attributes and outputting a decision value. This decision value can be either qualitative or quantitative, indicating a class or a number, respectively. A finalized decision tree representation shows a series of tests on the input attributes and the corresponding decision value to be returned. Russel & Norvig devised an example decision tree that can be used to answer the question "Should we wait for a table at this restaurant?" and this is shown in Figure 3. As seen, the tree attempts to make a decision as soon as possible, and places the most important factors higher up in the tree. For example, the number of patrons (None, Some, or Full), is often able to make a decision quickly.

2.1.2 Multiclass Classification

Multiclass classification is very similar to Binary Classification. Given a set of observations in a domain X, determine the value, Y, for each observation, where Y is some value that classifies the observation. This is a simple extension of Binary Classification, with Y assuming more than 2 values. Y does not necessarily have to assume a natural number as a value. In fact it is common for Y to denote more qualitative data. Consider the problem of identifying

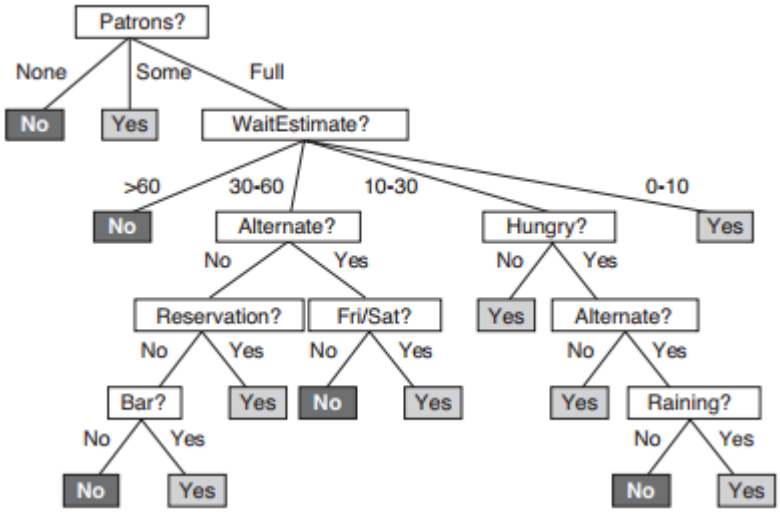

Figure 3: Boolean decision tree to determine whether to wait at a restaurant [9]

the language used in a certain spoken sentence. In this case, English may be associated with the value 1, Italian with 2, and Arabic with 3. There is also no limit to the number of classifiers. In this example, the spoken sentence will be one of hundreds of languages. The only limitations would be those imposed by the user for practical purposes. It is clearly possible, and common, to have a large number of potential classifiers [5].

Because of its similarity to Binary Classification, multiclass classification uses essentially the same algorithms to classify data. Decision trees, SVMs, and Perceptron algorithms can all be modified to classify data into more than 2 classes. In particular, there is a large body of literature exploring the modification of SVMs for multiclass classification. The foundation of all such methods and modifications to SVMs is to use K SVMs for a K-class data set [10]. Hsu & Lin performed very detailed analysis exploring the various modifications as well as their advantages and disadvantages [11]. They suggest that using a directed acyclic graph SVM (DAGSVM) is one of the more practical options.

2.1.3 Regression

Regression is commonly used in statistical analysis to measure the strength of a relationship between variables [9]. When used as a problem class in machine learning, Regression refers to the problem of, given a set of observations in a domain X, estimate the value Y, where Y is a real number. This is different from the problems of binary and multiclass classification. Rather than assign a categorical value Y to each observation, a real number Y is assigned instead. Many prediction problems fall into this category. Consider our example problem of predicting the temperature on a given day as illustrated in Table 2. Tomorrow's weather forecast can be summed up with one number, despite the large number of observations that factor into that prediction. It is likely that thousands upon thousands of observations led to the single numerical prediction. It is important

Table 2: Simple Regression example

Yesterday Temp	High/Low Pressure	Today's Temp
58	High	62.3
76.2	Low	70
66.4	High	68.1
35	Low	36
49	Low	? (A real number)

to keep in mind that this problem is not to classify each of those observations, but to use each one to derive a prediction.

Regression as a problem class in machine learning employs nearly identical techniques as in statistical analysis. One of the most common ways of determining a continuous variable from data is to simply find a Regression function from the training data and make predictions accordingly [7]. One technique that is more exclusive to the machine learning domain is to utilize a decision tree that returns continuous-numerical values rather than a qualitative classification. When used in this manner, the decision tree is referred to as a Regression tree. Essentially, the value returned by the Regression is the "mean of the target feature values of the instances from the training set that reached that node" [6]. Taking this value and subtracting it from the correct value gives us the error of the Regression tree.

2.2 Baseball Analytics

Due to its wealth of data and discrete nature, baseball readily lends itself to statistical analysis more than any other sport. Many books have been written on the subject, and baseball teams have, in recent years, prominently embraced data-driven and statistical analysis [12, 13, 14]. All the problem classes we've discussed can be applied to baseball. However, they are not often referred to as formally in much of the mainstream literature and culture.

There are many forms of statistical analysis applied to baseball that do not relate to machine learning. Even the simplest of statistics, such as batting average or a pitcher's win-loss record, are often useful in determining success of a player or team. Prior to Bill James's popularization of more complex analysis in the 1980s, these simple metrics served as the statistical foundation of baseball for decades [12]. Bill James is credited with popularizing the usage of *Sabermetrics*, although no precise definition for the term exists. In practice, *Sabermetrics* simply refers to any statistical analysis beyond the basic descriptive statistics such as batting average [1]. Advanced statistical analysis of baseball has become so popular that some of the largest baseball fan websites, such as Fan Graphs [4] and Baseball Savant [5], are dedicated to it.

One example of beyond-basic analysis is Bill James's Pythagorean expectation, which we present in Equation 1. This is still a relatively simple formula, but it goes beyond the basic win-loss ratio to calculate the expected number of wins for a team given their runs scored and runs allowed. The formula is as follows [15],

[4] `http://www.fangraphs.com`
[5] `https://baseballsavant.mlb.com`

$$ExpectedWinRatio = RunsScored^2/(RunsScored^2 + RunsAllowed^2) \quad (1)$$

This is simply one example of the typical statistical analysis employed for baseball. This Pythagorean expectation might be appropriate for a sports website or for amateur fans and analysts. The formula is relatively simple to understand and can be calculated quite easily. The machine learning analyses that we present are likely more appropriate for professional analysts, enthusiasts with a mathematical/scientific background, and academics interested in the field.

3 SLR Protocol

We followed the established systematic literature review (SLR) protocol as defined by Keele et al. [16]. Our main goal was for our methodology to be traceable and repeatable.

3.1 Research Questions

We used Pettigrew and Roberts's PICOC criteria to frame our research questions [17]:

- Population: Baseball Analysts (academic or industry), and others who might be interested in the intersection of machine learning and baseball analytics.
- Intervention: Machine learning techniques applied to baseball analytics. Specifically, machine learning techniques used for statistical analysis of performance.
- Comparison: Not applicable. We are interested in all techniques and classifying them as appropriate.
- Outcomes: We are looking for techniques that evaluate past performance and/or inform future decisions.
- Context: Approaches from both academic research and publicaly available industrial practice.

This allowed us to form the two research questions of this survey:

1. What are all the different ways machine learning has been applied to baseball?
2. What is the distribution of these applications across the machine learning problem classes?

3.2 Search Strategy

Our search strategy included all major online libraries relevant to this domain:

1. IEEExplore

2. ACM Digital Library
3. Google Scholar
4. Citeseer Library
5. Inspec
6. ScienceDirect
7. Ei Compendex
8. Journal of Sports Analytics
9. International Journal of Computer Science in Sport
10. Sloan Sports Conference
11. Fan Graphs Website
12. Baseball Savant

Our search strings involved finding the union of the word "Baseball" and a set of terms derived from the Bishop textbook on machine learning [7]. So, formally, our search string had the form of [*Baseball* & (Term1 OR Term2 ... OR TermN)]. The N terms included *machine learning, Binary Classification, Multiclass classification, Regression, supervised learning, unsupervised learning, novelty detection, statistical analysis, data analysis, data mining, prediction, analysis, models, evaluation, big data, inferring, inference, predict, stats, statistical, mining, data, model, modeling, neural net, markov, bayes, Bayesian, svm, support vector machine, hyperplane, expectation propagation, categorical, tuple, feature vector, feature, error rate, data cleaning, cross-validation, induction, regressor, decision tree, deep learning, reinforcement learning.*

3.3 Study Selection Criteria

Our selection of study criteria was as follows, with both being required,

- The study must address some aspect of the statistical analysis of baseball
- The study must utilize a machine learning approach

We further include:

- Any study that describes using machine learning techniques in any form to any level of baseball

And we exclude:

- Any study that does not focus on the statistical analysis of performance. For example, highlight extraction from raw video or business analysis.

3.4 Study Selection Procedure

Kaan Koseler performed the study extraction and collection using the search terms defined above. Inclusion/Exclusion was performed through collaboration between Kaan Koseler and Matthew Stephan. Disputes to this effect were resolved through discussion.

3.5 Study Quality Assessment Procedure

Due to exploratory nature of this survey, we decided to disregard study quality. We used several public, non-academic articles which were not peer reviewed. These were noted as such in our report.

3.6 Data extraction strategy

The data extraction for each approach and article involved manually determining the problem class being explored and what the approach is being used for. We examined each of the papers in detail to extract out the results of their work.

3.7 Synthesis of the extracted data

This was not a formal meta-analysis in that we were not evaluating the success of some particular intervention. Rather, we were simply compiling an exhaustive list of studies and categorizing them by problem class (Binary Classification, Multiclass Classification, and Regression). The synthesis consisted of determining the problem class(es) explored by the article and categorizing it accordingly.

3.8 Dissemination Strategy

We will publish a longer draft version of this document as a technical report in the Miami University Technical Repository. We will submit refined versions to journals/conferences. .

3.9 Results of Protocol

In total, we found 145 articles using our search terms and strategy. Of those 145, 32 articles met our inclusion criteria and 117 were excluded. This high exclusion rate is explained by both the inclusion and the exclusion criteria. Many of the articles employed statistical analysis of performance without using a machine learning technique. This is in large part due to the search term "Regression" having two slightly different meanings between traditional statistical analysis and machine learning. There are many applications of a Regression problem which involve analyzing the correlations between variables, but this is not a machine learning problem as no predictions are being made. There were also several studies found that used machine learning but did not apply such techniques to analysis of performance, but rather to analyzing ticket sales or other financial matters.

4 Machine Learning Applied To Baseball

Machine learning's predictive power has led to its use in baseball for both practical and research applications. Machine learning analysis improves with increasing numbers of observations. This is readily illustrated when one considers extremes. Suppose the goal is to predict if a pitcher's next pitch will be a fastball or not. If there are only one or two observations of the pitcher's past pitches, it will be nearly impossible to accurately predict the next pitch with

significant accuracy. However, if there are one hundred thousand observations, the accuracy of the prediction will be fairly high. Major League Baseball has the opportunity for a relatively large number of observations featuring seasons of 162 games per season with 30 teams. Additionally, the technology at this level is much more advanced than the amateur levels, providing access to more types and granularity of data. Thus, machine learning is very viable for professional baseball, as it is a good candidate for having accurate and strong predictive power.

In the following section, we organize our presentation by considering each of the Machine Learning problem classes one at a time as they apply to baseball. This is done to explicate the distribution of techniques across the problem classes, as expressed in our second research question. For each technique, we first summarize how each has been employed thus far in the field by presenting applications and any related/extend versions of their use. We include notable and illustrative examples in detail. Secondly, we present some of our insights on how each respective technique can be leveraged in this field in the future.

4.1 Binary Classification

4.1.1 Existing Work

Consider our earlier example of predicting whether a pitcher's next pitch will be a fastball or not. This is a Binary Classification problem, in that there are two classes: fastball and non-fastball. Previous research has demonstrated excellent predictive improvements when using machine learning for this exact problem [18, 19]. Ganeshapillai and Guttag used a linear Support Vector Machine (SVM) classifier to classify pitches based on data from the 2008 season and predict the pitches of the 2009 season. SVMs are a type of supervised learning algorithm [18]. The 2008 pitching data was feedback contained labeled examples of the type of pitch that was thrown by a pitcher. For Ganeshapillai and Guttag, they achieved a significant performance improvement over a naive classifier. The naive classifier can be thought of as a simple Bayes classification based on probability. In other words, if a pitcher in 2008 used a fastball greater than 50% of the time, the naive classifier would predict that every pitch in 2009 would be a fastball. The model Ganeshapillai and Guttag created was able to correctly predict the next pitch 70% of the time, whereas the naive classifier was able to correctly predict the next pitch 59% of the time. Thus, they achieved an 18% improvement (59*1.18) over the naive classifier. Some of the more representative examples from their paper are presented in Figure 4. The I column indicates percentage performance improvement of the SVM (A_o) as compared to the naive classifier (A_n). As demonstrated, it is difficult to improve upon pitchers who overwhelmingly utilize one pitch, such as Mariano Rivera. Rather, this Support Vector Machine (SVM) approach holds more promise when a pitcher is more unpredictable by human standards, such as Andy Sonnanstine. This type of analysis using a SVM is probably the most widely used method for Binary Classification in general. It is relatively simple to implement and provides good performance [9].

Hoang et al. also studied the problem of classifying pitches into fastball and non-fastball categories [19, 20]. Their work examined the approach to the problem itself by comparing the different Binary Classification algorithms by

	Name	ERA (2009)	Pitches		Accuracy		I
			Training	Test	A_o	A_n	
Greatest Improvement	Andy Sonnanstine	6.77	3125	1675	32%	8%	311%
	Brian Bannister	4.73	3074	2475	47%	16%	196%
	Miguel Batista	4.04	2022	1169	72%	36%	100%
	Scott Feldman	4.08	2356	3145	65%	35%	85%
	Rafael Perez	7.31	1098	694	73%	40%	79%
	Francisco Rodriguez	3.71	1109	1154	71%	44%	62%
	Kyle McClellan	3.38	1128	1060	76%	49%	53%
	Nick Blackburn	4.03	2830	3154	62%	41%	52%
Highest Accuracy	Mariano Rivera	1.76	911	1202	94%	92%	1%
	Tim Wakefield	4.58	2693	1998	93%	89%	4%
	Mark DiFelice	3.66	318	742	92%	92%	0%
	Bartolo Colon	4.19	550	974	90%	90%	0%
	Roy Corcoran	6.16	1066	304	89%	88%	1%
	Matt Thornton	2.74	1013	1095	88%	88%	0%
	Aaron Cook	4.16	2996	2448	86%	86%	0%
	Jesus Colome	7.59	1131	334	84%	79%	6%
Least Accuracy	Andy Sonnanstine	6.77	3125	1675	32%	8%	311%
	Brian Bannister	4.72	3074	2475	47%	16%	196%
	Chad Durbin	4.39	1371	1290	56%	49%	16%
	Francisco Cordero	2.16	1176	1012	58%	39%	48%
	Jason Jennings	4.13	506	1025	59%	51%	14%
	Braden Looper	5.22	3168	3214	59%	48%	22%
	Darren Oliver	2.71	1052	1187	59%	47%	27%
	Cliff Lee	3.22	3235	4068	59%	59%	0%

Figure 4: Salient examples from Ganeshapillai & Guttag [18]

prediction accuracy. Figure 5 summarizes their results. Their four algorithms included k-nearest neighbors (kNN), Support Vector Machine with linear kernel (SVM-L), Support Vector Machine with Gaussian kernel (SVM-G), and Linear Discriminant Analysis (LDA). Interestingly, the SVM approach fared the worst among the approaches they employed. The authors suggest that LDA be used. LDA, compared against SVM, achieved roughly 78% accuracy. A key difference between LDA and SVM is that LDA is fundamentally Bayesian in nature. It uses Bayes Theorem to determine the probability that an observation belongs to a given class. This can be written as $Pr(Y = i|X = x)$, denoting the probability that observation $X = x$ belongs to class i. In contrast, an SVM will simply classify the observation with no regard to probability of belonging to a certain class. Another way to think of this difference is that LDA will produce a generative model for new data, whereas SVM will not [7]. This work was also repeated in Hamilton et al. [21]. Hamilton et al. focused their work on improving feature selection and achieved a modest improvement in prediction accuracy when focusing on dominant features only.

Soto Valero conducted baseball research that compared the effectiveness of different algorithms in both the Regression and Binary Classification domain [22]. The problem that he explored was predicting the outcome of a baseball game, win versus lose, for all 30 teams in the MLB using 10 years of historical data as training data. Four algorithms were compared against one another: 1) k-nearest neighbors, referred to as 1-NN in their work; 2) artificial neural networks, referred to as MLP due to the specific Multi-Layer Perceptron

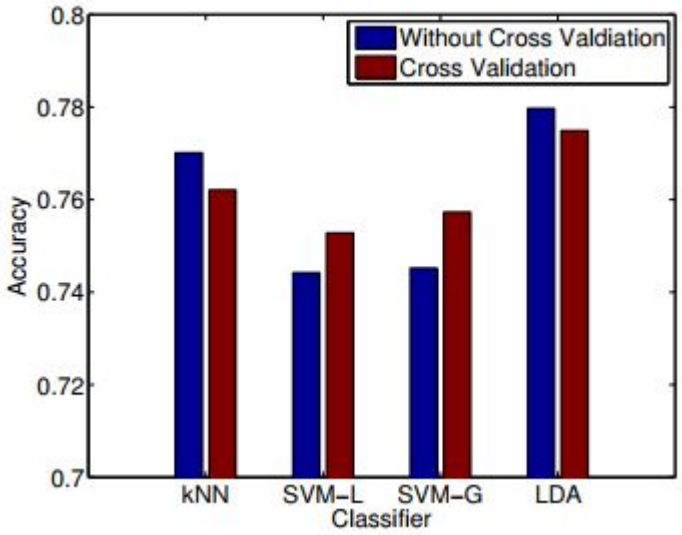

Figure 5: Comparison of Binary Classification Approaches [19]

version used; 3) Decision Trees , implemented in Weka using REPTree; and 4) SVM, using the SMO implementation. We present their results in Figure 6. They indicate that the SVM approach was the most successful, with prediction accuracy of just under 60%.

Work by Jang et al. proposed an approach that attempted to predict whether a given player in South Korea would be asked to join the South Korean national baseball team [23]. They collected past data on nine players that were on the national team, and used five candidate players as test data. A kNN algorithm was used, but no results were presented. This work was more a proposal into exploring the feasibility of the kNN algorithm.

4.1.2 Discussion: Potential Applications

There are many potential applications for Binary Classification in baseball. Simple examples include classifying a match up between two teams as a win or a loss, deciding if a player will bunt, and classifying whether a certain team will choose to intentionally walk the player in a situation. In the first example, observations might consist of a vector of players and their individual performances using statistics such as on-base percentage or batting average. For the bunting and intentional walk examples, situational data like outs, the game score, and a player or manager's predilection for doing so can be considered by analysts.

Considering the results demonstrated in the literature, we recommend starting with either an SVM approach or an LDA. Both algorithms have concrete evidence demonstrating good results in prediction accuracy. Although Hoang et al. suggest using LDA, the improvement shown in Figure 5 is only marginal when compared against the SVM approach. Each analyst should determine through experimentation which algorithm is best for the problem domain and their interests.

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

Figure 6: Comparison of Algorithms used to Predict Win-Loss [22]

4.2 Multiclass Classification

4.2.1 Existing Work

A simple example of Multiclass classification in baseball is classifying pitches beyond the simple binary of "Fastball" and "Not Fastball". Instead analysts can classify non-fastball pitches by pitch type, such as curve balls, change ups, or sliders. Even the "Fastball" category can be further subdivided into cutters, two-seam fastballs, four-seam fastballs, et cetera. An analyst may want to classify pitches into one of more than two different classes.

Sidle's work during their graduate studies is the most expansive. They detail three different methods to classify pitches into seven different types. Sidle employs "Linear Discriminant Analysis, Support Vector Machines, and bagged random forests of Classification Trees" to classify pitches [24]. They achieved improvements in prediction accuracy over a simple naive classifier. Their results demonstrated that the forest of Classification Trees was the superior method, with Linear Discriminant Analysis second, and Support Vector Machines behind. This is illustrated in Figure 7, where the last four columns represent the accuracy in percentage and the first two columns represent the training and test data set sizes, respectively. 100CT indicates a forest of 100 classification trees. Sidle notes that Linear Discriminant Analysis is both more efficient and more

Starters						
Pitcher	Training	Testing	Naive	LDA	SVM	100 CT
R.A. Dickey	7786	2385	88.89	89.31	89.94	89.64
Kevin Gausman	2492	1386	88.31	88.89	88.67	88.82
Lance Lynn	7497	2091	86.90	86.85	87.33	87.33
Taijuan Walker	1490	1905	83.94	84.72	83.20	85.20
Bartolo Colon	6293	1944	83.95	83.44	83.90	84.57
Jake Odorizzi	4262	1831	85.53	80.17	80.17	84.27
Alfredo Simon	5080	2185	81.33	79.82	79.91	82.11
Ubaldo Jiminez	6077	2153	80.12	79.01	78.36	80.03
James Paxton	1809	823	71.81	78.25	77.64	78.86
Juan Nicasio	4446	762	75.33	78.48	75.85	78.35
Relievers						
Pitcher	Training	Testing	Naive	LDA	SVM	100 CT
Koji Uehara	2072	472	99.30	99.30	99.30	99.30
Sam Freeman	1053	476	92.86	92.44	87.61	93.91
Zach Putnam	1063	490	87.78	90.02	84.52	91.04
Jake McGee	2334	384	89.84	90.63	91.15	90.36
Brad Brach	1840	800	86.63	90.00	89.75	89.75
Tony Cigrani	3069	468	88.03	88.03	89.32	89.74
Jonathan Papelbon	2067	649	83.20	89.06	88.44	89.52
Zach Britton	1937	674	31.16	87.98	88.43	89.02
Kenley Jansen	2513	588	85.88	86.39	86.54	87.07
Jeremy Jeffress	974	760	76.45	81.71	81.71	84.74

Figure 7: Comparison of Algorithms used by Sidle [24]

consistent than the forest of Classification Trees. They measured efficiency in computation time. Sidle found that using a naive guess, 51% of pitches thrown by starters were accurately predicted, compared to 57% accuracy for relievers [24]. The major issue with pitch prediction is that there are different types of pitchers, primarily broken down into starters and relievers. This same issue was encountered in the Binary Classification techniques also, as Ganeshapillai and Guttag discovered pitchers for whom there was little or no improvement, like Mariano Rivera [18]. This difference is likely due to the need for starting pitchers to utilize a larger arsenal of pitches, whereas relievers might be more likely to rely on a smaller number of pitches. Starting pitchers play more innings and throw more pitches than relievers, so a starter with only a few pitch types will quickly be exploited by the opposing team. Because of this difference in pitch arsenal between starters and relievers, models produced by different methods may show consistency differences in prediction. There are even differences between pitchers of the same type. Even among starters, some are easier to predict than others. Sidle shows that using a naive classifier, we can predict 88% of R.A. Dickey's pitches. However, Juan Nicasio is harder to pin down with their pitches being predicted with 75% accuracy.

Bock et al.,[25] performed similar work with Multiclass Classification of pitch

Machine Learning Algorithm	Model Accuracy	Attribute (1)	Attribute (2)
Linear Kernel SVM	57.1%	Fielding Percentage	Batting Average
Quadratic Kernel SVM	60.7%	WHIP	ERA
Cubic Kernel SVM	67.9%	Double Plays turned	Wins
Gaussian Kernel RBF	69%	SLG	Double plays turned

Figure 8: Comparison of Different SVM Kernels [29]

type. The main focus of their work was on using these predictions to build a model of the pitcher's long-term performance as we describe here. The authors used both multinomial logistic Regression and Support Vector Machine algorithms to train their models. They derived an overall prediction accuracy of 74.5%, which was better than Sidle's prediction accuracy of roughly 65% across all three methods. From these predictions, Bock et al. derived a pitch sequence predictability measure referred to as the "predictability index". They further used this index in a linear Regression analysis in an attempt to predict long-term ERA, but the Regression analysis revealed that a pitcher's predictability index was not correlated with long-term ERA.

Other work by Attarian et al. used data from the PITCHf/x system to classify pitches thrown by certain pitchers into different pitch types [26, 27]. These types were not explicitly stated in their paper, but were based on different characteristics like spin rate and velocity of the pitch. They used a kNN algorithm and compared it against a naive Bayes classifier. Much like other work we present, the kNN algorithm achieved an improvement over the naive Bayes classifier. In this case an average of 4% improvement in prediction accuracy was observed by the authors. Another analysis they performed was using LDA to reduce the number of features for predicting pitch type. Using LDA in this manner reduced the features down to 4 dominant predictors: spin rate, spin direction, break angle, and start speed. However, using these 4 predictors provided only an 1.68% improvement in prediction accuracy.

Ishii used clustering algorithms to determine undervalued players and classify them based on pitch type and repertoire [28]. They used both k-means clustering and hierarchical clustering in their analysis, seeking to find players whose ERA was higher than their cluster ERA. The cluster ERA represents an average ERA for a player of that skill level. Players who fit this criteria were deemed to be undervalued. This identifies them as players that a baseball general manager might sign as a bargain. Ishii found no difference in classification based on pitch type or repertoire, and both were equally effective in determining undervalued players when using clustering algorithms.

Tolbert and Trafalis used a SVM to determine the winners and losers for the American League Championship Series (ALCS), the National League Championship Series (NLCS), and the World Series in Major League Baseball [29]. The main component of their analysis was using a different kernel for the SVM and assessing the accuracy of the resulting prediction. They also examined the features that were best in making a prediction. We summarize the results of their work in Figure 8, showing the most predictive attributes for each SVM. The SVM using a Gaussian Radial Basis Function (RBF) kernel was the most accurate overall.

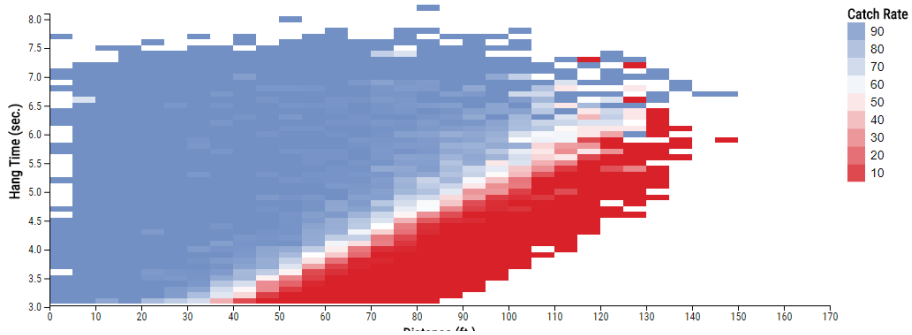

Figure 9: Catch Probability Based on Hang-Time [30]

One public and popular source that employs Multiclass Classification is the website BaseballSavant[6]. They use "hang-time", that is, the time that a batted ball spends in the air, and ball travel distance to categorize the "catchability" of a hit ball [30]. This classification is based on the probability of a defender catching the ball. The classes themselves are explicitly delineated. For example,the $50-75\%$ catch probability category. Figure 9 demonstrates this concept and the relationship between catch probability and hang time.

4.2.2 Discussion: Potential Applications

As is the case for Binary Classification, the SVM approach is well-studied and effective for Multiclass Classification. One possible approach that emulates the one taken by Ishii [28] is to group players into undervalued, correctly valued, and overvalued groups. This would be of significant use to general managers and other personnel in a baseball team's front office. Rather than using traditional metrics, analysts could use batting average or advanced metrics to cluster batters and determine which "value" group they belong to. This can be combined with a salary analysis to allow baseball teams to determine players' worth and to assist in decisions to sign or cut players from the team. It can also be used to evaluate the performance of young players in the minor leagues and determine if they should be promoted to the major league team. For example, ascertaining "success" factors for minor league players that will translate to the major leagues, such as the exit velocity of their hits.

Ishii has demonstrated that clustering using the k-means algorithm is effective for identifying undervalued players [28]. This is what we would recommend as a starting point for others looking to cluster players. Attarian et al. demonstrated that the kNN algorithm is effective for classifying pitch types, and combining this with LDA for feature selection is even more effective [27]. Baseball teams might utilize a similar approach when preparing for their opponents and studying their pitching habits.

[6] `https://baseballsavant.mlb.com/`

4.3 Regression

4.3.1 Existing Work

One area of research we discovered measures and shows a decline in batting average over a player's career [31]. That is, the authors are using age as a feature vector, x, to derive a batting average, y. The results concur with common knowledge that athletes tend to perform at a lower level as they age.

Healey [32] uses the log5 model to assess the probability of a strikeout given a specific match up between a pitcher and a batter. The log5 model is an analogous calculation to that of the Elo rating, which is used in chess to predict the probability of a win in a match between two players. In addition to a match between pitcher and batter, log5 has commonly been used to evaluate a match between two whole teams to estimate winning percentages. We present its canonical usage in Equations 2 and 3. Let A be the winning percentage of team A and B be the winning percentage of team B. If want to know the probability, P, that team A will win against team B, we can use log5 as follows,

$$P = \frac{A - A * B}{A + B - 2 * A * B} \tag{2}$$

Healey modifies this basic formula to model the probability of a strikeout E^* as follows:

$$E^* = \frac{(BP)/L}{(BP)/L + (1 - B)(1 - P)/(1 - L)} \tag{3}$$

where B is the batter's strikeout rate, P is the pitcher's strikeout rate, and L is the average league strikeout rate. Healey goes on to incorporate ground ball rates into the model as well. These ground ball rates were studied more thoroughly in later work by Healey [33], along with an investigation into their impact on the E^* formula described above.

Barnes and Bjarnadottir used Regression models to assess free agent performance, and similar to Ishii [28], used these models to identify undervalued and overvalued players [34]. The models they used were linear Regression, linear Regression with feature selection, Regression trees, and gradient-boosted trees. They determined linear Regression with feature selection models had the greatest potential for identifying highly overvalued or highly undervalued players. Feature selection indicated that WAR, wins above replacement, was the best statistic for predicting future performance. This was measured by calculating a surplus value, indicating performance greater than that predicted by the Regression model. Figure 10 shows the pitchers with the highest predicted surplus value.

Das and Das took a different approach with their work, blending psychology and machine learning to analyze which aspects of a ball in flight contribute most to a fielder's ability to catch it successfully [35]. They began with a working hypothesis that the elevation angle of the ball from the fielder's perspective is the most important contributor to catching success and evaluated their hypothesis by building a neural network model that could learn to "catch" balls. By continually feeding velocity and elevation angle of balls in air, the neural network was able to predict the proper coordinates to position itself to catch the ball. Their final results indicated that towards the end the ball's flight, its velocity becomes more important to catch probability than elevation angle, which is still a large contributor.

Player	Pos	Year	Age	Team	WAR	Predicted WAR	Next WAR	New average yearly salary	Performance value	Market value	Excess value
Chris Sale	SP	2012	23	CHW	5.9	5.5	6.9	6.6	20.4	8.7	11.7
Barry Zito	SP	2001	23	OAK	4.5	4.8	7.2	3.1	17.8	7.1	10.7
David Wells	SP	2003	40	NYY	4.3	3.5	2.6	1.6	13.1	2.9	10.2
Tim Lincecum	SP	2009	25	SFG	7.5	6.3	3.7	12.5	23.2	13	10.2
Joakim Soria	CL	2008	24	KCR	3.7	3.4	2.7	3.2	16	6	10
James Shields	SP	2007	25	TBD	5.5	4.7	3.8	3.1	17.3	7.4	9.9
Johan Santana	SP	2004	25	MIN	8.6	5.9	7.2	12.1	21.9	12.3	9.6
Clayton Kershaw	SP	2011	23	LAD	6.5	6.4	6.2	9.8	23.4	14.3	9.1
Aaron Harang	SP	2006	28	CIN	5.2	4.9	6	10.5	18	9.1	8.9
Josh Johnson	SP	2009	25	FLA	6.6	4.3	7.2	10.6	15.8	7.9	8

Figure 10: Pitchers with Highest Predicted Surplus Value [34]

$$\begin{aligned}\text{OPG} = 0.30&\left(\frac{\text{BA}-0.263}{0.027}\right)+0.35\left(\frac{\text{OBP}-0.326}{0.036}\right)\\&+0.37\left(\frac{\text{SLG}-0.380}{0.064}\right)+0.39\left(\frac{\text{OPS}-0.706}{0.094}\right)\\&+0.38\left(\frac{\text{TA}-0.651}{0.129}\right)+0.32\left(\frac{\text{ISO}-0.118}{0.049}\right)\\&+0.34\left(\frac{\text{SECA}-0.222}{0.072}\right)+0.38\left(\frac{\text{RC27}-4.569}{1.322}\right).\end{aligned}$$

Figure 11: The Offensive Player Grade Metric [37]

Everman created his own statistic referred to as Calculated Aggregate Value (CAV) to predict the winner of a playoff series [36]. We present their algorithm in Equation 4. In evaluating a matchup, the team with the higher CAV was predicted to win. Everman assessed the 2004 MLB playoff season and states only that the CAV made the correct prediction "in nearly every instance" [36]. The author posits that this novel CAV statistic can be used as an excellent predictor in future research.

$$\begin{aligned}CalculatedAggregateValue = AdjustedProduction * WinningPercentage\\ + FieldingAverage * WinningPercentage \quad (4)\end{aligned}$$

Tung developed their own statistic that attempts to measure a player's performance. They refer to it as the Offensive Player Grade (OPG) [37]. This metric measures only a player's offensive performance and ignores their defensive statistics. We present their formula for calculating this metric in Figure 11 Tung used Principal Components Analysis (PCA) to develop this metric, analyzing the various offensive statistics of a player to determine which ones are relevant their appropriate weights. Then, Tung proceeded to use k-means to cluster these players into groups based on their OPG score. Baseball analysts might find this metric of use when assessing a player's offensive value.

Freiman demonstrated the feasibility of using Random Forests to predict a player's election to the Baseball Hall of Fame [38]. Results indicated that Freiman achieved 75% prediction accuracy using the Random Forests. Only 1% of the players who were actually elected were predicted not to be elected by Freiman's approach. They state the most important individual statistic to

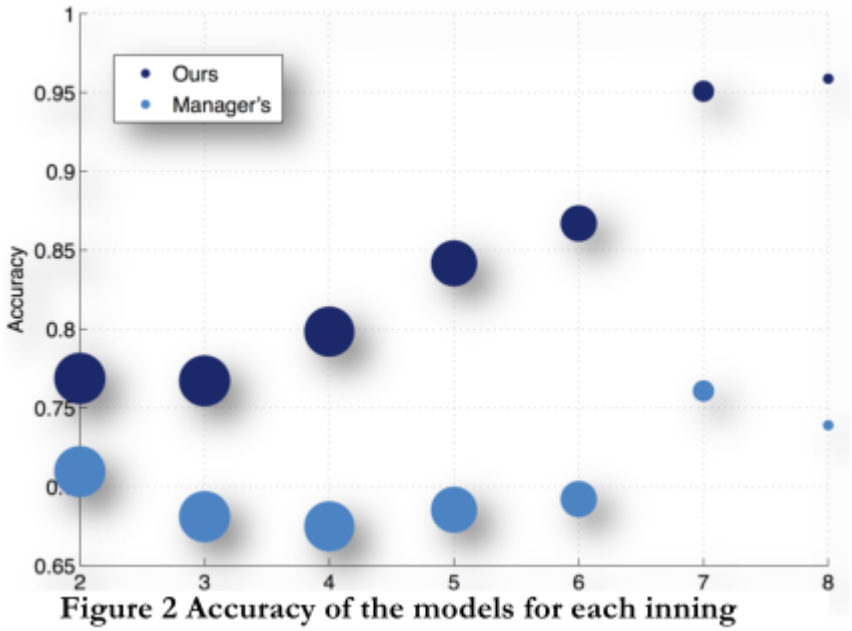

Figure 12: Prediction Accuracy when Pulling a Pitcher [39]

predict Hall of Fame election was the number of home runs throughout the player's career.

Ganeshapillai and Guttag, who also investigated the Binary Classification problem of predicting the next pitch, developed a model to determine when a starting pitcher should be pulled from the game. They first trained a manager model that would pull the pitcher based on data that would be available to a manager during the game and actual decisions made by major league managers. The authors then built a regularized linear Regression model to predict whether a pitcher, if not pulled, would surrender a run in the next inning [39]. This model disagreed with the manager model a surprising 48% of the time, achieving an impressive improvement in accuracy over the manager model. This accuracy was measured by whether or not the pitcher surrendered a run in the next inning. Figure 12 shows this improvement over the manager model.

Herrlin explored fantasy baseball roster optimization [40]. They used a Bayesian approach to build models for both pitchers and batters. These models were used to build Regression trees that would be able to predict outcomes for the player throughout the rest of the season. They also explored batting order optimization using the results returned by the Regression trees. There was no single statistic used for this optimization, but rather different Regression trees modeling different statistics such as batting average or ERA.

Huddleston used Bayesian machine learning to predict future performance in fantasy baseball, but in this case used the single statistic of fantasy points to compare players [41]. Another difference between this analysis and Herrlin's is that Huddleston creates three different models that vary in their treatment of hitters and pitchers. The first model does not differentiate between the two, the second model distinguishes between hitters and pitchers only, and the third

model further distinguishes starting and relief pitchers. These models are all trained by using prior distributions of points scored from previous seasons as training data. The results indicate the second model has the greatest fit to the data. Additionally, Huddleston finds that pitchers should always be preferred over hitters as, on average, they score more fantasy points.

Jensen et al. developed a detail Bayesian model to assess the evolution of hitting performance over a player's career [42]. They utilize several different techniques in building their model, including 1) hidden Markov Chains to model movement between "elite" and "non-elite" status, and 2) Gibbs sampling [43] to estimate posterior distributions of home run rates. Figure 13 shows the difference in home run rates between combinations of elite and non-elite designated hitters and shortstops. Beyond the common sense trend of declining performance with age, the authors also show that "elite" players have steeper declines than "non-elite" players. Similar work has also been done by Stevens in attempting to model a pitcher's strikeouts and walks as they age over time [44]. Using a logistic Regression model of the past 100 hundred years of historical pitcher data, the curves in Figure 14 show that Elite players suffer from greater declines in performance than non-elite players, but tend to maintain excellent performance until well into their 30s. The curves also show that peak performance tends to occur around age 25.

Jiang and Zhang used Bayesian methods to predict a player's batting average in the 2006 MLB season [45]. They used batting averages from the 2005 season as their training data. Their main goal was to show the feasibility of Empirical Bayes over a simpler least-squares predictor [46]. The results indicated that Empirical Bayes "may capture a great portion of the effects of missing covariables in the linear model" [45]. This leads to the recommendation that analysts consider using empirical Bayesian methods rather than a simple linear Regression least squares model.

Another Bayesian approach was devised by Yang and Swartz to calculate the probability of a team winning a certain game [47]. They combine home field advantage with past performance, batting ability, and starting pitchers of a match up in a two-stage Bayesian model. The first stage assumes that the probability of a team winning is a "random sample from a beta distribution with parameters based on the relative strength variable and the home field advantage variable." The second stage is a random sample from a Bernoulli distribution of the first stage's probability. This is combined with Gibbs sampling from a Monte-Carlo Markov Chain to make predictions.

Lyle uses a variety of techniques including SVMs, artificial neural networks, and model trees to predict several different offensive statistics [48]. These are the typical statistics used to evaluate a player's offensive prowess, such as runs, RBIs, hits, triples, and doubles. Lyle compared these techniques against existing baseball prediction systems such as the Player Empirical Comparison and Optimization Test Algorithm (PECOTA) [49], which uses a nearest neighbor search comparing players to other players, and the Szymborski Projection System (ZiPS) [50]. The results showed that Lyle's predictors were only able to outperform the existing PECOTA and ZiPS systems on the triples statistic. On all other statistics, the existing systems were superior. Both PECOTA and ZiPS are proprietary systems.

Panda approaches the problem of using penalized Regression models to reduce the number of features required to make predictions [51]. Beginning with

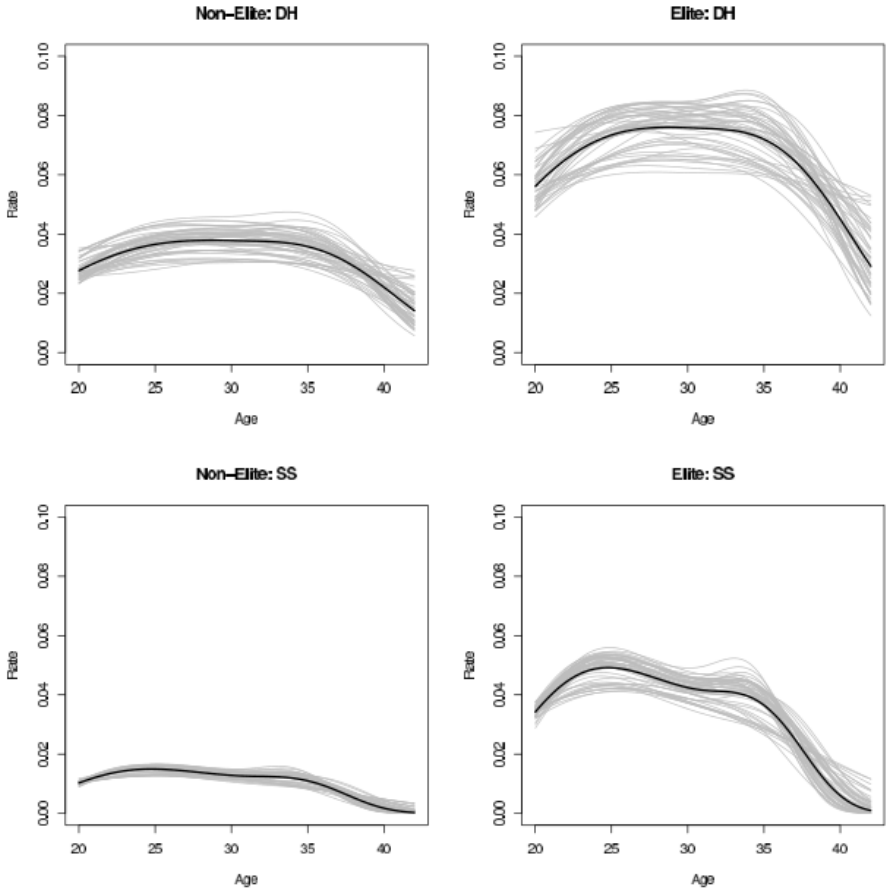

Figure 13: Home Run Rates Over Time for Elite and Non-Elite Players [42]

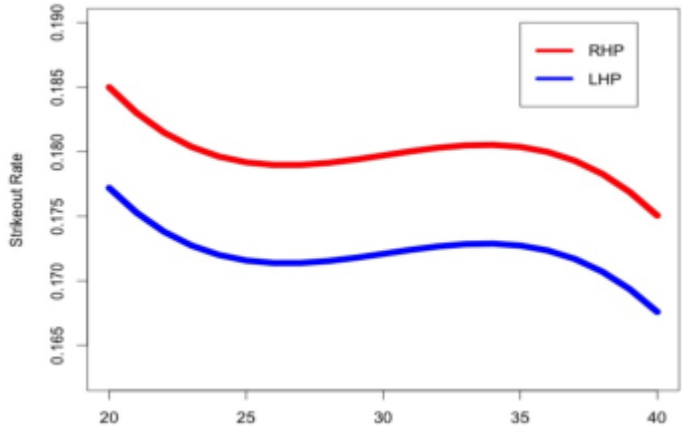

Figure 14: Strikeout Rate Over Pitchers Careers[44]

a total pool of 31 different offensive metrics and 5 defensive metrics, Panda showed that using penalized Regression models results in reducing these metrics down to only seven offensive metrics and two defensive metrics. In essence, these metrics are what "distinguish a player over time" and were determined by analyzing their "signal". They indicate what percentage of players differed from the overall mean. Those metrics with a high signal were deemed to be worthy of inclusion in the model.

An unfinished proposal by Reeves posits that the k-NN algorithm can be used to predict player performance through clustering [52]. This is essentially the same procedure used by the PECOTA system, but Reeves gives two major differences. Rather than assess only a three-year window of a player's career, Reeves assesses their entire career. The difference is that only one performance statistic will be generated, compared against the seven stat lines from PECOTA, each with their own confidence interval.

There was one public source that used Bayesian Regression to develop a model that would assess a player's batting ability: the LingPipe blog [53]. It was based primarily on a player's batting average, with the author measuring batting ability as a number between 0 and 1. Greater weight was given to consistent performance over increasing numbers of batting opportunities, or "at bats". For example, the author shows that using their statistic would classify a player with 40 hits and 100 at bats, a .400 batting average, as inferior to a player with 175 hits out of 500 at bats, a .350 batting average. Another public resource authored by Sidran describes importing data from EVN files into Microsoft Excel and running some simple statistical analysis on a pitcher's "score". This "score" metric is a basic evaluation of the pitcher and may be a useful tool for analysts looking for a simple but effective pitching metric. Figure 15 shows the scoring system for this metric. This article is an unfinished proposal as it relates to the domain of machine learning. Sidran posits that this metric can be used to predict when a pitcher should be pulled from the game, taking the form of a probability that a pitcher will "falter" by falling below a certain running score threshold. This decision would be based on past data indicating the point at which a pitcher has faltered in the past.

4.3.2 Discussion: Potential Applications

There are plentiful Regression problems in baseball analytics. For instance, an analyst might be interested in predicting a player's batting average for the season and use data collected from past seasons as training data to make that prediction. A variety of algorithms can be applied. In the literature we presented various forms of Bayesian Inference were used often and achieved good performance. An analyst might also utilize a Linear Regression model, which is easier to implement, but this might result in decreased prediction accuracy.

Analysts might also wish to predict a pitcher's ERA, again using data collected from past seasons to train a Bayesian model that can be used to make predictions. Future research could employ applying Artificial Neural Networks for such tasks. Given the increasing popularity of deep learning libraries like TensorFlow and Torch, these might offer better performance and ease of implementation than the techniques employed in the articles we presented. Although few examples of neural networks were found during our survey, it is impossible to ignore their current domination of the machine learning field as a whole.

Event:	Value added to pitcher's score
Ball	-1
Strike	+1
Walk	-1
Single	-1
Double	-2
Triple	-3
Home Run	-4
Stolen Base	-1
Ball Put In Play	+1
Foul Ball	+1

Figure 15: Score Metric to Evaluate Pitcher Performance [54]

5 Summary and Discussion

Baseball Analytics is a large and ever-growing field that has been strongly incorporated into the professional leagues, particularly Major League Baseball. Machine learning, although not a new field, has recently seen tremendous growth in research interest both in the academic and public domains. The problem classes' application in baseball that we explored in the literature are Binary Classification, Multiclass classification, and Regression. Although there are other machine learning problem classes [5], they are of limited utility to baseball analysis.

In total, we found 5 articles exploring Binary Classification, 8 articles for Multiclass classification, and 19 articles for Regression. These 32 articles were drawn from a pool of 145 candidate articles, of which 115 were excluded for either failing to meet inclusion criteria or for meeting the exclusion criteria. We summarize the articles from our systematic literature review and their problem classes in Table 3.

In reviewing the literature, we noticed several algorithms that were frequently used. In particular, Support Vector Machines and Bayesian inference were popular approaches. SVMs are used for classification problems, both binary and Multiclass. Bayesian inference can be used for any problem class. Of the articles that we reviewed, Bayesian inference was the most often used for Regression tasks, which are themselves the most common in the literature. Some of the articles we reviewed made use of existing machine learning software like WEKA or R to run their analyses. However, many researchers chose to implement their algorithms manually. This demonstrates the relative ease of implementation for many of these approaches and that future researchers or analysts do not need to limit themselves to working with existing software. We present a ranking of the popularity of the approaches in Table 4. As shown, SVM and KNN approaches were used most often, each appearing at least 25% of the time. Despite the current domination of Artificial Neural Networks in the machine learning literature, they were only used in 9% of the articles meet-

ing our inclusion/exclusion criteria. We anticipate that this will change in the coming years.

6 Conclusion

Considering the recent growth in interest in machine learning and the popularity of baseball, there are bound to be future researchers who study this intersection of baseball analytics and machine learning. The machine learning approach is a natural fit for so-called "sabermetrics", as it allows the analyst to glean insights from the rich data generated by baseball and leverage it to a competitive and professional advantage. In our review, we discovered that Support Vector Machines were the most popular method of classification, while Bayesian Inference mixed with Linear Regression was the most popular method for Regression tasks. It should be noted that while Support Vector Machines can only be used for classification tasks, Bayesian Inference can be used for both classification and Regression, although the articles discussed in this report only used Bayesian Inference for Regression tasks. We anticipate that this will change in the coming years. There is currently dominance of neural networks in the machine learning literature and a proliferation of libraries like TensorFlow and Torch that allow users to quickly build and train neural networks. Neural networks also have the advantage of being useful for both classification and Regression tasks. It is our prediction that baseball analytic research will catch up, and begin employing neural networks for analysis.

Our hope is that this report will serve as a go-to resource for those interested in learning about the intersection of machine learning and baseball. We also hope to help facilitate those pursuing further research and baseball analysis.

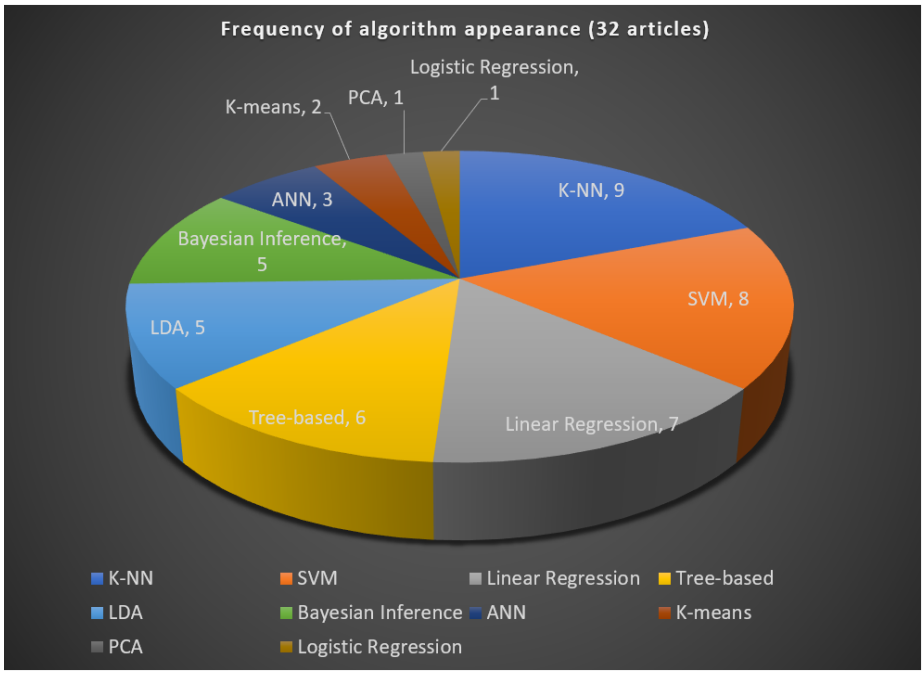

Figure 16: Frequency of algorithm approaches

Table 3: Reviewed Articles Categorized by Problem Class

Problem Class	Title	Author(s)
Binary Classification	Predicting the Next Pitch	Ganeshapillai & Guttag [18]
	Supervised learning in Baseball Pitch Prediction and Hepatitis C Diagnosis	Hoang [20]
	A Dynamic Feature Selection Based LDA Approach to Baseball Pitch Prediction	Hoang et al. [19]
	Analyzing and Predicting Patterns in Baseball Data using Machine Learning Techniques	Jang et al. [23]
	Predicting Win-Loss outcomes in MLB regular season games	Soto Valero [22]
Multiclass classification	Baseball pitch classification: a Bayesian method and dimension reduction investigation	Attarian et al. [27]
	A comparison of feature selection and classification algorithms in identifying baseball pitches	Attarian et al. [26]
	Applying machine learning techniques to baseball pitch prediction	Hamilton et al. [21]
	Winning Baseball Through Data Mining.	Swan & Scime [55]
	Using Machine Learning Algorithms to Identify Undervalued Baseball Players	Ishii [28]
	Using Multi-Class Machine Learning Methods to Predict Major League Baseball Pitches.	Sidle [24]
	Predicting Major League Baseball Championship Winners through Data Mining	Tolbert & Trafalis [29]
Regression	Great expectations: An analysis of major league baseball free agent performance	Barnes & Bjarnadottir [34]
	Catching a baseball: a reinforcement learning perspective using a neural network	Das & Das [35]
	Analyzing Baseball Statistics Using Data Mining	Everman [36]
	Predicting Major League Baseball Playoff Chances Through Multiple Linear Regression	Firsick [56]
	Using random forests and simulated annealing to predict probabilities of election to the Baseball Hall of Fame	Freiman [38]
	A Data-driven Method for In-game Decision Making in MLB	Ganeshapillai & Guttag [39]
	A Bayesian Approach to Markov-Chain Baseball Analysis	Hammons [57]
	Forecasting MLB performane utilizing a Bayesian approach in order to optimize a fantasy baseball draft	Herrlin [40]
	Hitters vs. Pitchers: A Comparison of Fantasy Baseball Player Performances Using Hierarchical Bayesian Models	Huddleston [41]
	Hierarchical Bayesian modeling of hitting performance in baseball	Jensen et al. [42]
	Empirical Bayes in-season prediction of baseball batting averages	Jiang et al. [45]
	Baseball prediction using ensemble learning	Lyle [48]
	Penalized Regression Models for Major League Baseball Metrics	Panda [51]
	Regression planes to improve the pythagorean percentage	Moy [58]
	Major League Baseball Performance Prediction	Reeves [52]
	A Method of Analyzing a Baseball Pitcher's Performance Based on Statistical Data Mining	Sidran [54]
	Bayesian Statistics and Baseball	Stevens [44]
	Data Mining Career Batting Performances in Baseball	Tung [37]
	A two-stage Bayesian model for predicting winners in major league baseball	Yang & Swartz [47]

Table 4: Approaches Ranked By their Prevalence in the Literature

Approach	Included Articles Using Approach
K-nearest neighbors	9/32 = 28.1%
Support Vector Machine	8/32 = 25%
Linear Regression	7/32 = 21.8%
Tree-based Methods	6/32 = 18.75%
Linear Discriminant Analysis	5/32 = 15.6%
Bayesian Inference	5/32 = 15.6%
Artificial Neural Network	3/32 = 9.4%
K-means	2/32 = 6.25%
Principal Component Analysis	1/32 = 3.13%
Logistic Regression	1/32 = 3.13%

References

[1] Gabriel B Costa, Michael R Huber, and John T Saccoman. *Understanding sabermetrics: An introduction to the science of baseball statistics.* McFarland, 2007.

[2] Bill James. *The Bill James Baseball Abstract 1987.* Ballantine Books, 1987.

[3] Travis Sawchik. *Big Data Baseball: Math, Miracles, and the End of a 20-Year Losing Streak.* Flatiron, 2015.

[4] Robert P Schumaker, Osama K Solieman, and Hsinchun Chen. *Sports data mining methodology.* Springer, 2010.

[5] Alex Smola and S.V.N Vishwanathan. *Introduction to Machine Learning.* Cambridge University Press, 2008.

[6] John D Kelleher, Brian Mac Namee, and Aoife D'Arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* MIT Press, 2015.

[7] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[8] Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

[9] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2 edition, 2003.

[10] Eddy Mayoraz and Ethem Alpaydin. Support vector machines for multiclass classification. *Engineering Applications of Bio-Inspired Artificial Neural Networks*, pages 833–842, 1999.

[11] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[12] Michael Lewis. *Moneyball: The art of winning an unfair game*. WW Norton & Company, 2004.

[13] Gabriel B Costa, Michael R Huber, and John T Saccoman. *Reasoning with Sabermetrics: Applying Statistical Science to Baseball's Tough Questions*. McFarland, 2012.

[14] Benjamin Baumer and Andrew Zimbalist. *The sabermetric revolution: Assessing the growth of analytics in baseball*. University of Pennsylvania Press, 2013.

[15] S. J. Miller. A Derivation of the Pythagorean Won-Loss Formula in Baseball. *ArXiv Mathematics e-prints*, September 2005.

[16] Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn, 2007.

[17] Mark Petticrew and Helen Roberts. *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.

[18] Gartheeban Ganeshapillai and John Guttag. Predicting the next pitch. In *Sloan Sports Analytics Conference*, 2012.

[19] Phuong Hoang, Michael Hamilton, Joseph Murray, Corey Stafford, and Hien Tran. A dynamic feature selection based lda approach to baseball pitch prediction. In *Trends and Applications in Knowledge Discovery and Data Mining*, pages 125–137. Springer, 2015.

[20] Phuong Hoang. *Supervised Learning in Baseball Pitch Prediction and Hepatitis C Diagnosis*. North Carolina State University, 2015.

[21] Michael Hamilton, Phuong Hoang, Lori Layne, Joseph Murray, David Padget, Corey Stafford, and Hien Tran. Applying machine learning techniques to baseball pitch prediction. In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods*, pages 520–527. SCITEPRESS-Science and Technology Publications, Lda, 2014.

[22] C Soto Valero. Predicting win-loss outcomes in mlb regular season games–a comparative study using data mining methods. *International Journal of Computer Science in Sport*, 15(2):91–112, 2016.

[23] Wu-In Jang, Aziz Nasridinov, and Young-Ho Park. Analyzing and predicting patterns in baseball data using machine learning techniques. 2014.

[24] Glenn Daniel Sidle. *Using Multi-Class Machine Learning Methods to Predict Major League Baseball Pitches*. PhD thesis, North Carolina State University, 2017.

[25] Joel R Bock. Pitch sequence complexity and long-term pitcher performance. *Sports*, 3(1):40–55, 2015.

[26] A Attarian, G Danis, J Gronsbell, G Iervolino, and H Tran. A comparison of feature selection and classification algorithms in identifying baseball pitches. In *International MultiConference of Engineers and Computer Scientists*, pages 263–268, 2013.

[27] A Attarian, G Danis, J Gronsbell, G Iervolino, L Layne, D Padgett, and H Tran. Baseball pitch classification: a bayesian method and dimension reduction investigation. In *IAENG Transactions on Engineering Sciences: Special Issue of the International MultiConference of Engineers and Computer Scientists 2013 and World Congress on Engineering 2013*, page 393. CRC Press, 2014.

[28] Tatsuya Ishii. Using machine learning algorithms to identify undervalued baseball players.

[29] Brandon Tolbert and Theodore Trafalis. Predicting major league baseball championship winners through data mining.

[30] Statcast catch rates. [Online; https://baseballsavant.mlb.com/statcast_catch_probability].

[31] Mark Fichman and Michael A Fichman. From darwin to the diamond: How baseball and billy beane arrived at moneyball. 2012.

[32] Glenn Healey. Modeling the probability of a strikeout for a batter/pitcher matchup. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2415–2423, 2015.

[33] Glenn Healey. Matchup models for the probability of a ground ball and a ground ball hit. *Journal of Sports Analytics*, (Preprint):1–16.

[34] Sean L Barnes and Margrét V Bjarnadóttir. Great expectations: An analysis of major league baseball free agent performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(5):295–309, 2016.

[35] Rajarshi Das and Sreerupa Das. Catching a baseball: a reinforcement learning perspective using a neural network. In *AAAI*, pages 688–693, 1994.

[36] Brad Everman. Analyzing baseball statistics using data mining.

[37] David D Tung. Data mining career batting performances in baseball. *Journal of Data*.

[38] Michael H Freiman. Using random forests and simulated annealing to predict probabilities of election to the baseball hall of fame. *Journal of Quantitative Analysis in Sports*, 6(2), 2010.

[39] Gartheeban Ganeshapillai and John Guttag. A data-driven method for in-game decision making in mlb. 2014.

[40] Daniel Luke Herrlin. *Forecasting MLB performane utilizing a Bayesian approach in order to optimize a fantasy baseball draft.* PhD thesis, San Diego State University, 2015.

[41] Scott D Huddleston. Hitters vs. pitchers: A comparison of fantasy baseball player performances using hierarchical bayesian models. 2012.

[42] Shane T Jensen, Blakeley B McShane, Abraham J Wyner, et al. Hierarchical bayesian modeling of hitting performance in baseball. *Bayesian Analysis*, 4(4):631–652, 2009.

[43] Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[44] Guy Stevens. *Bayesian Statistics and Baseball.* PhD thesis, PhD thesis, Pomona College, 2013.

[45] Wenhua Jiang, Cun-Hui Zhang, et al. Empirical bayes in-season prediction of baseball batting averages. In *Borrowing Strength: Theory Powering Applications–A Festschrift for Lawrence D. Brown*, pages 263–273. Institute of Mathematical Statistics, 2010.

[46] Rolland L Hardy. Least squares prediction. *Photogrammetric Engineering and Remote Sensing*, 43(4), 1977.

[47] Tae Young Yang and Tim Swartz. A two-stage bayesian model for predicting winners in major league baseball. *Journal of Data Science*, 2(1):61–73, 2004.

[48] Arlo Lyle. *Baseball prediction using ensemble learning.* PhD thesis, uga, 2007.

[49] Nate Silver. Introducing pecota. *Baseball Prospectus*, 2003:507–514, 2003.

[50] Baseball Prospectus. Baseball think factory. *Blogs: High-impact Strategies-What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*, page 61, 2012.

[51] Mushimie Lona Panda. *Penalized Regression Models for Major League Baseball Metrics.* PhD thesis, University of Georgia, 2014.

[52] Jason Reeves. Major league baseball performance prediction.

[53] Bayesian estimators for the beta-binomial model of batting ability. [Online; https://lingpipe-blog.com/2009/09/23/bayesian-estimators-for-the-beta-binomial-model-of-batting-ability/].

[54] D Ezra Sidran. A method of analyzing a baseball pitcher's performance based on statistical data mining, 2005.

[55] Gregory Swan and Anthony Scime. Winning baseball through data mining. In *DMIN*, pages 151–157, 2010.

[56] Zachary Firsick. *Predicting Major League Baseball Playoff Outcomes Through Multiple Linear Regression.* PhD thesis, University of South Dakota, 2013.

[57] Christopher Hammons. *A Bayesian Approach to Markov Chain Baseball Analysis.* PhD thesis, Georgetown College, 2006.

[58] D Moy. Regression planes to improve the pythagorean percentage: A regression model using common baseball statistics to project offensive and defensive efficiency. *Statistics, University of California Berkeley, Berkeley, CA, MS Thesis*, 2006.