Using Machine Learning to Predict MLB Hitter Performance

Joe Germino

GitHub: https://github.com/jgermino/Advanced-Data-Analytics

Abstract

Baseball projection systems have become a central part of online baseball analysis and help inform nearly every decision made by Major League Baseball (MLB) Front Offices. This paper explores various machine learning techniques - specifically Decision Trees, K-Nearest Neighbors, Linear Regression, and stochastic Gradient Boosting Regression – in an attempt to improve on existing publicly available projection systems in predicting a single all-encompassing hitting statistic, Weighted on Base Average (wOBA). Validation models were built using a variety of datasets allowing for advanced analysis into the strengths and weaknesses of each model as well as the underlying data. Testing reveals the potential of using Linear Regression and Gradient Boosting Regression at predicting player performance while Decision Trees and K-Nearest Neighbors were relatively weak. Meanwhile, the limited training data proved to be a challenge for each of these models. Data imputation methods appear to help mitigate these issues though there are signs of potential bias which may be limiting its overall effectiveness. In total, these models performed favorably against the generally-accepted baseline displaying encouraging signs for the potential of combining these techniques with more rigorous statistical analysis in the future.

Introduction

Since the early 2000s, projection systems have served as the foundation of modern-day baseball analysis. In addition to the publicly available systems, all 30 MLB front offices are believed to operate their own proprietary system. These systems drive player analysis and assist front offices in much of their decision-making including free agent and trade valuation. Any marginal improvement to these projection systems could provide a team with a competitive advantage. The goal of this paper is to use common Machine Learning techniques to gather insights into their applicability to baseball and identify possible methods for improving on current systems.

Related Work

There exist many different publicly available projection systems most prominently PECOTA, ZiPS, and Steamer. While the exact specifics of these systems are not public, there is enough information available online to build a general idea of their methodologies and generate an understanding of what makes a

good projection system. As Dan Szymborski, creator of ZiPS described, most projection systems have two major components: estimating the baseline expectation for a player and then estimating how that changes based on similar players past performance (Szymborski, 2020). Alternatively, PECOTA maintains a database of roughly 20,000 major league batter seasons and creates a similarity score between a player's past performance and its historical player seasons. It then uses these comparable players to estimate a probability distribution of the player's upcoming performance (PECOTA, n.d.). Additionally, PECOTA uses three years of player performance in their comparisons while ZiPS requires four. While each of the various approaches in this paper are unique, similar to PECOTA many of the models will rely on trying to identify similarities between past player seasons and predicting based on this information.

Additionally, Tatsuya Ishii at Stanford used k-means clustering to identify similar pitchers in order to find undervalued players and generated some positive results (Ishii, 2016). While the focus and methodology of their research does not directly relate to this report, it is encouraging to see other successful research utilizing machine learning and demonstrating the possibility of grouping players based on some underlying characteristics.

Overview

Goal

The goal of this project is to use to K-Nearest Neighbors Regression, Decision Tree Regression, Linear Regression, and Gradient Boosting Regression to predict a qualified hitter's Weighted on Base Average in the 2019 season given the player's performance in prior years.

Expectation

Prior to training any models, the expectation is that Linear Regression will be the best performing model. Gradient Boosting Regression and K-Nearest Neighbors will hopefully also perform favorably in comparison to Marcels and provide useful insights but seem unlikely to match Linear Regression. Finally, Decision Trees are their own seem unlikely to provide useful results but may prove useful as a data imputation method given their wide-usage in industry.

Baseline

Statistician Tom Tango developed the Marcel the Monkey Forecasting System (Marcels) as "the most basic forecasting system you can have, that uses as little intelligence as possible" (Tango T. , 2012). The Marcels are calculated by taking a weighted average of the player's previous 3 seasons, regressing the value towards the mean, and applying an aging curve (Tango T. , 2004). This system is designed to be the bare minimum of what a projection system is able to forecast. Any system that performs worse or on par with the Marcels is generally considered to have added complexity without accuracy and should usually be avoided. For this reason, the Marcels will be used as a baseline against which the models are tested. Marcel projections were provided courtesy of Baseball Reference.

Data

The training data was downloaded from Fangraphs and consists of all qualified hitter seasons from 1930 to present. Fangraphs maintains a free online database of hundreds of statistics for every player dating

back to 1871 and has strong data integrity with minimal missing or erroneous values. A hitter is deemed to be qualified in any season in which they average 3.1 plate appearances per team game[1]. As will be discussed in further detail below, most of the features used to train the models were rate statistics. By limiting the training data to qualified seasons, the rate statistics used should be fairly stable and more indicative of true talent level. Baseball, as with any human activity, is inherently random so the qualified threshold was used as an arbitrary cut-off to prevent small sample noise from dominating the model. The initial training set consisted of 10,009 player seasons. However, because the model is trying to predict a player's wOBA in the following year, the training data must be limited to players with two consecutive seasons in the data set. Therefore, the final training set only contained 6,707 player seasons.

Data Normalization

Given changes in various league and era trends, statistics from 1930 are not directly comparable to statistics from 2019. As an example, league-wide K% has steadily increased from 7.1% in 1930 to 23.4% in the 2020 season. To combat the effects the changing landscape would have on the models, all data was first normalized. The data was converted to z-scores which were calculated using the mean and standard deviation of the training set grouped by season. Much baseball research uses more sophisticated statistical techniques for league and era adjustments. However, the z-scores proved to be sufficient for these purposes.

Prediction

The models were built to predict a player's Weighted on Base Average (wOBA). wOBA is one of the most popular all-encompassing offensive statistics. It was developed by baseball statisticians Tom Tango, Mitchel Lichtman, and Andrew Dolphin. wOBA measures a hitter's overall offensive value based on the relative values of different offensive events, such as walks, singles, home runs (Tango, Lichtman, & Dolphin, 2006).

Feature Selection

The full list of features used to train the models as well as their definitions are included in the Appendix. The features used in fitting the model were handpicked either because prior research suggests they are strong indicators of future performance or they are frequently cited statistics in modern-day baseball analysis. Additionally, some Statcast metrics that were not trackable prior to recent technological innovations were also included as features as a matter of interest. The features selected were mostly rate statistics. Since the training sets were limited to qualified seasons, the rate stats should have stabilized enough and offer a better indication of performance than counting statistics. Many of the selected features correlate strongly with each other. This is intentional and was handled through Principal Component Analysis.

[1] "Qualified" is an official MLB designation used for end of season award eligibility

Training Data

The training data was divided into 14 different training sets looking at different time periods and utilizing different imputation methods. Specifically, models were trained using training sets looking at players performance in the previous season, the previous two seasons, and the previous three seasons. As previously mentioned, some of the features selected were not available dating back to 1930 for technological reasons or otherwise. Additionally, there were many cases of player's not qualifying in consecutive seasons within the training data resulting in missing features. For each of the training sets, copies of the training set were fit with no imputation, imputation through linear regression, and imputation through decision trees. For each of the imputation methods, Principal Component Analysis was used such that the number of dimensions chosen explained 95% of the variance. The full explanation of each training set is provided in the Appendix.

Parameter Tuning

For K-Nearest Neighbors and Decision Trees, an important factor in the model is the selection of the k and depth parameters, respectively. A different value was chosen for each training set. To determine these values each validation set was plotted with the root mean squared error (RMSE) on the y-axis and the parameter on the x-axis with a line for each PCA dimension. An example of these graphs can be found in the Appendix. From there, the parameter was selected based on the "elbow" in the graphs – the point at which the curve begins to flatten. Notably, not all dimensions had the same elbow and not every graph had a clear point to be considered the elbow. As a result, this step required a mixture of objective and subjective analysis.

Principal Component Analysis

To determine the number of dimensions for each model and each training set a similar procedure was followed as above. In this case, the number of dimensions after PCA reduction was plotted on the x-axis instead of the parameter. A line was included on the graph for each of the 4 model types and the appropriate number of dimensions for the training set again was determined by the elbow in the graphs. An example of these graphs can be found in the Appendix. As before, some subjective analysis was required to pick the appropriate number of dimensions.

Hyperparameter Tuning

For Gradient Boosting Regression, Gridsearch was used to find the optimal Learning Rate and number of estimators at each dimension. Once the PCA value was fixed, the model was rerun multiple times on the validation data to get estimates for the optimal values. From there, an approximate average was chosen for the final model.

Validation Results

The final results using the validation sets are below. Note that the values are the RMSE of the predicted z-scores. The blue shading corresponds to lower numbers while the red shading higher.

Z-Scores Validation Testing	1-Year Basic	1-Year Advanced	1-Year All	2-Year All
K-Nearest Neighbors	0.843	0.874	0.840	1.016
Decision Tree	0.864	1.057	1.098	0.920
Linear Regression	0.831	0.831	0.877	0.887
Gradient Boosting Regression	0.864	0.880	0.908	0.930

No imputation

Z-Scores Validation Testing	1-Year DT Imputed	2-Year DT Partial Imputed	2-Year DT Imputed	3-Year DT Partial Imputed	3-Year DT Imputed
K-Nearest Neighbors	0.924	0.983	0.913	1.064	0.925
Decision Tree	0.892	0.919	0.862	1.052	0.952
Linear Regression	0.849	0.891	0.869	1.019	0.858
Gradient Boosting Regression	0.864	0.898	0.865	1.034	0.902

Imputed with Decision Trees

Z-Scores Validation Testing	1-Year LR Imputed	2-Year LR Partial Imputed	2-Year LR Imputed	3-Year LR Partial Imputed	3-Year LR Imputed
K-Nearest Neighbors	0.893	0.921	0.845	1.026	0.876
Decision Tree	0.869	0.913	0.822	0.991	0.874
Linear Regression	0.831	0.829	0.789	0.913	0.782
Gradient Boosting Regression	0.863	0.837	0.782	0.939	0.788

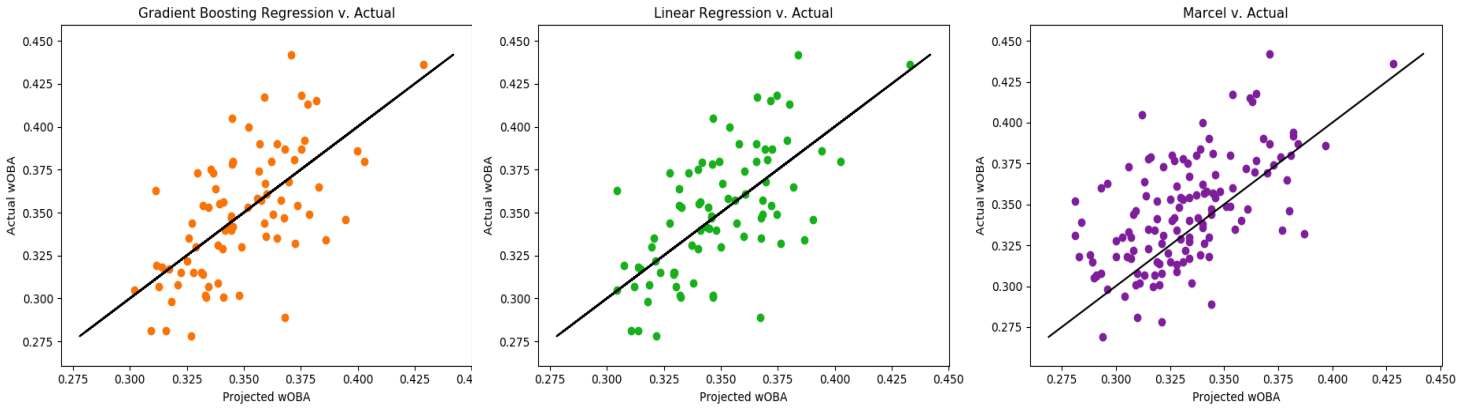
Imputed with Linear Regression

The final validation results indicate that the best training set to use was the data containing two years of information on each player using linear regression to impute the missing data. Additionally, the Gradient Boosting Regression and Linear Regression models performed significantly better across the board than the Decision Tree and K-Nearest Neighbors models. While Gradient Boosting Regression performed slightly better than Linear Regression in the optimal case, Linear Regression generally was the superior model. Combining this with the prior expectation that Linear Regression would outperform Gradient Boosting Regression, Linear Regression with 2-years imputed data appears to be the best model.

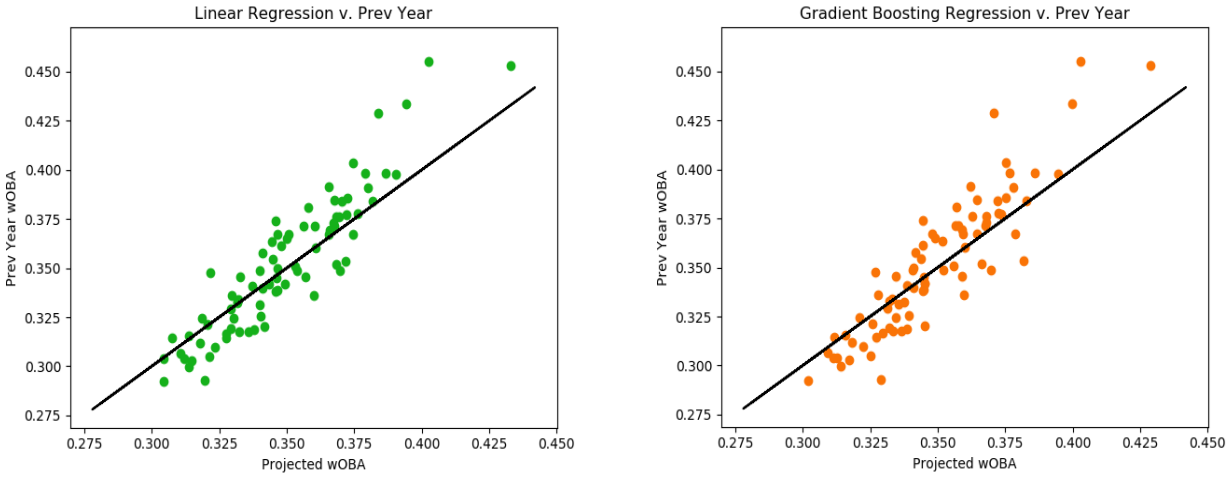
Test Results

The test results for all models using the 2-Year Linear Regression Imputed training data is below. The predicted values were converted from z-scores to wOBA and the RMSEs are shown in terms of wOBA.

Test Results	RMSE	R^2
Linear Regression	0.0281	0.407
Gradient Boosting Regression	0.0285	0.389
K-Nearest Neighbors	0.0296	0.340
Decision Tree	0.0298	0.333
Marcels	0.0311	0.346



Encouragingly, all four models performed similarly to Marcel while Gradient Boosting Regression and Linear Regression both appear to have been significant improvements. This appears to demonstrate as a proof of concept that these methods could indeed have a place in furthering baseball projection systems' accuracy.

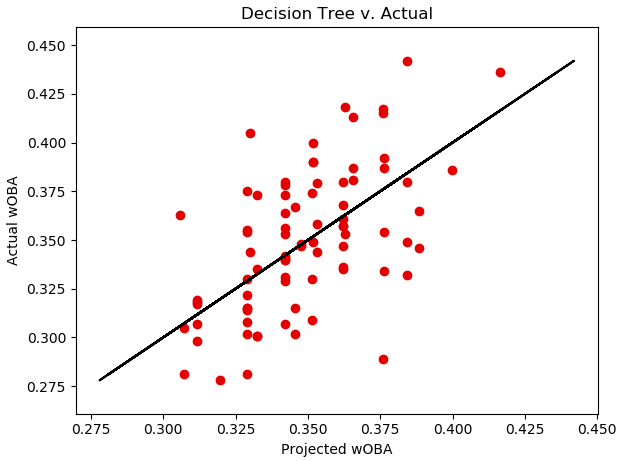


The above graphs show the models' predicted wOBA versus the prior year's wOBA. While the models' predictions are similar to the prior year's performance, they also at times predicted significant improvement or decline. This is generally a good sign that the models may be picking up indicators that a naïve prediction would otherwise miss providing further evidence of their usefulness.

Further Analysis

Decision Tree Regression

Decision Tree Regression generally performed poorly relative to the Linear Regression and Gradient Boosting Regression at predicting wOBA. However, while still performing slightly worse than Linear Regression, using Decision Trees to impute data significantly improved the performance of the models. The explanation for why this may be the case seems intuitive when we examine the test results of the final Decision Tree model:



As the image shows, there generally appears to be a reasonably strong correlation between the Decision Tree's prediction and the actual wOBA value. That is to say that the Decision Tree appears to be decent at grouping players into buckets of expected performance. However, where decision trees suffer is the lack of precision in their estimates. Using the heuristics described above, the optimal depth for this particular training set was set to 5. While this shallow depth prevents overfitting, it also results in many of the vertical stacks present in the graph. Other models were better able to recognize the smaller differences in these groupings and separate the players more precisely.

K-Nearest Neighbors Regression

K-Nearest Neighbors Regression disappointingly performed very similarly to Decision Trees overall. At the onset of this project, the prior belief was that K-Nearest Neighbors would outperform Decision Trees and fare adequately against Linear Regression. The reason for this underperformance appears to be directly related to the decision to limit the training set to qualified player seasons and the effect this had on the size of the training data. The below tables show the RMSE of K-Nearest Neighbors on each training set against the number of samples in that set. As above, the blue shading corresponds to lower numbers while the red shading higher.

K-Nearest Neighbors	1-Year Basic	1-Year Advanced	1-Year All	2-Year All	1-Year DT Imputed	2-Year DT Partial Imputed	2-Year DT Imputed
Size of Training Set	6707	1459	180	59	6707	4750	6707
RMSE	0.843	0.874	0.840	1.016	0.924	0.983	0.913

K-Nearest Neighbors	3-Year DT Partial Imputed	3-Year DT Imputed	1-Year LR Imputed	2-Year LR Partial Imputed	2-Year LR Imputed	3-Year LR Partial Imputed	3-Year LR Imputed
Size of Training Set	3465	6707	6707	4750	6707	3465	6707
RMSE	1.064	0.925	0.893	0.921	0.845	1.026	0.876

In the above chart, it is evident that there is an inverse correlation between the size of the training set and the overall error of the model. This makes sense as K-Nearest Neighbors typically performs best in scenarios with large training sets. Additionally, K-Nearest Neighbors is subject to the curse of dimensionality and, even with PCA, some of these training sets still required 10 or more dimensions to fit the model.

One oddity in this relationship is the performance of the "1-Year Advanced" and "1-Year All" training sets. Despite being two of the three smallest training sets, they were two of the four best models. The

explanation for this is not evident at present. It seems likely this is a fluke of the data, but it may be worth exploring further in the future.

Ultimately, K-Nearest Neighbors still performed on par with Marcels despite a relatively small training set. It might still be a strong candidate for future projection systems but further research with larger training sets will need to be conducted.

Gradient Boosting Regression

Gradient Boosting Regression is the first of the two successful models tested. The model was built using stochastic gradient boosting with a subsample size of .5. The final model used a learning rate of .05 with 150 estimators. The Gradient Boosting Regression results are most interesting when compared directly to the results of the lone decision tree model. The general idea behind boosting methods at large is to create an ensemble of weak prediction models which when combined create a single strong predictor. In this particular case, 150 decision trees were built using half the training data each time.

As discussed above, it seems counter-intuitive that decision trees would be a successful predictor and there is evidence to suggest that, on their own, they are not. Additionally, the above model was fit using the full training set which, as previously discussed, appears to have caused issues in training for its small size. Given these facts, the prior belief was that Gradient Boosting Regression could perform well but would be outperformed by Linear Regression. Instead, this project provides an interesting case study into how effective boosting models can be in seemingly counter-intuitive situations. Going forward, it will be interesting to see if Gradient Boosting Regression can keep pace with, or even surpass, Linear Regression under improved methodologies. These results seemingly indicate that Gradient Boosting Regression is at least a viable candidate for usage in future projection systems.

Linear Regression

The prior belief beginning this project was Linear Regression would outperform the other models in predicting wOBA. The end results appear to support this prior. The relative success of Linear Regression seems explainable via the bias-variance tradeoff. Both K-Nearest Neighbors and Decision Trees are high variance models subject to overfitting. These models rely heavily on the training data and do not generalize on the data it has not seen. This is not the case with Linear Regression which is a high bias and low variance model. Given that baseball performance is inherently random, it makes sense that a high bias model which does not overfit to the training data would be a strong predictor.

Survivor Bias

As mentioned above, one of the main issues with the models appeared to be the lack of training data. Given this, one lingering question may be why was there no attempt to impute the y value for players who do not have 2 consecutive seasons? This would have increased the training sample by nearly 50% and might have trained better models. However, this is not advisable due to survivor bias.

In order to impute data, the missing data must be assumed to be randomly distributed. Otherwise, the model will be working with a biased training sample. However, that is not necessarily the case here. While there are many reasons why a player may not have 2 consecutive qualified seasons, including circumstances beyond their control such as injury, the most common explanation is due to poor performance or expected poor performance. Whether it is related to age, poor play leading to reduced

playing time, or some other unknown factors, if teams have reason to believe a player has declined in ability, that player is less likely to achieve the "qualified" threshold. As a matter of fact, we know from past research from Mitchel Lichtman (Lichtman, 2009) and Jonathan Judge (Judge, 2020) that a similar survivor bias exists when calculating MLB aging curves. They each have separately presented strong evidence of its effects on their models. Therefore, it is likely that players who have two consecutive qualified seasons are simply better than players without it and the missing data cannot be assumed to be randomly distributed.

Bias in Imputed Data

This, however, leads to another interesting question: how certain is it that the data that was imputed is not biased in some other way? It is important to first separate the different data imputations that occurred. The majority of missing data were the new stats in the "Advanced" and "All" training sets for which tracking technology did not exist until recently. For these stats, because the determining factor of whether or not the data exists is Season which theoretically should not affect the underlying values, there does not seem to be any reason to believe the missing data is not random.

The other, more relevant data imputation occurred when the training sets were expanded to include two- and three-years' worth of data. In this case, it is plausible that the data was biased. Similar to the argument for survivor bias, there may be a "newcomer" bias. Whereas before, the issue was players whose performance significantly declined, the new issue is players whose performance significantly improved.

Looking at the results, there may be indications of this bias effecting the outcomes. Specifically, imputing did significantly improve the accuracy of the models in the two-year case. However, the three-year case, while still performing better than the models with no imputation, performed worse than the two-year models. One theory for why this may be the case relates again to the relative lack of training data. Theoretically, it is possible that the imputed data has some bias to it but that the added value of the extra data points outweighs the added error of the bias. This could also explain why the three-year data performed worse. Since the three-year data was imputed partially using the two-year data, the bias that exists may be compounded to the point that the additional error is beginning to detract from the overall value added. Considering that past research has indicated a minimum of three-year period is typically ideal, this theory could explain why the three-year results in this project are subpar.

Overall, it seems plausible that the underlying data is biased as discussed but the overall effect is small enough year-to-year to not meaningfully affect the projections. But as the lookback period increases and the model trains itself on this data, the bias becomes amplified and begins to have a more meaningful effect on the predicted values. This could be an area for improvement in future research.

Conclusion and Next Steps

Overall, there are encouraging signs in the data provided here to indicate that some of these Machine Learning techniques have potential in future projection systems. However, there remain several areas for improvement and further research before that becomes the case. As repeatedly mentioned throughout this paper, the biggest hindrance to the models' success appeared to be related to the lack of training data. There could be opportunity for future research working with a larger data set to repeat the experiments conducted herein. The simplest way to accomplish this would be to remove the qualified threshold on player seasons. To avoid issues with the model being dominated by small sample seasons with unreliable data, it may be possible to weight the samples based on a playing time metric such as Plate Appearances.

Alternatively, it may be interesting to mix these models with existing techniques such as regression to the mean and explicit aging curves. For example, instead of using the raw data provided by Fangraphs, regress the numbers and add an aging curve and then train a model without knowledge of the player's age. This technique could also help expand the data set beyond qualified players.

Finally, as mentioned, the potential bias in the imputed data is worth further exploration to more precisely identify and hopefully mitigate the issue.

Description of Effort

Overall, I really enjoyed working on this project. While the initial vision was to see how far I can push K-Nearest Neighbors Regression in building a projection system, it soon evolved into a much more interesting case study comparing four different model types on real world data. I spent much of my time reading on the specifics of these different models and familiarizing myself with their strengths and weaknesses. I felt it was important for me to establish some priors of the expected performance of each model so I had a frame of reference for analyzing them after the fact. This helped identify things which I considered to be surprising either good or bad. I dedicated a good amount of time to combing through the final validation results and trying to identify any trends in the results and this ended up being the most interesting part for me even more so than the actual test results.

The majority of my time was spent either coding or developing the methodology for each step in the process. All of the code on the GitHub repository was written by me. I worked extensively with pandas and scikit-learn, and while I have prior experience with both of them, given the scale of this project, it was a great opportunity to learn new methods and become more familiar with some less common features. There were also several tests which I began working on but had to scrap as I felt it no longer lined up with the vision of the project. The biggest one involved feature engineering and analysis. I also tried working with a more sophisticated normalization method than the z-scores but was unable to gain access to the data I needed to finalize the test results in time to make it into the report.

Fortunately, I have extensive experience working with baseball data and am usually up to date on baseball research as it comes out. This made it much easier for me to deal with many of the intricacies of the data and quickly look up relevant research when necessary. That familiarity also helped when it came time to address many of the issues associated with real world data such as missing data and

potential biases. It was much easier to think through these problems than it otherwise would have been with completely unfamiliar data. On the flip side, it was interesting to go through the analysis process and see what information I could glean about baseball modeling specifically and to identify potential research areas for myself in the future.

Appendix

Training Features

Feature	Definition	Year Available
Age	Player's age as of June 30 in the specified season	All
PA	Plate Appearances	All
AVG	Batting Average	All
OBP	On Base Percentage	All
SLG	Slugging Percentage	All
OPS	On Base Plus Slugging	All
ISO	Isolated Power	All
BABIP	Batting Average on Balls in Play	All
BB%	Walks per Plate Appearance	All
K%	Strikeouts per Plate Appearance	All
LD%	Line Drivers per Ball in Play	2002
GB%	Ground Balls per Ball in Play	2002
FB%	Fly Balls per Ball in Play	2002
GB/FB	Ground Balls per Fly Ball	2002
IFFB%	Infield Fly Balls per Fly Balls	2002
HR/FB	Home Runs per Fly Ball	2002
O-Swing%	Swings at Pitches Outside Strike Zone per pitches seen outside Strike Zone	2002
Z-Swing%	Swings at Pitches Inside Strike Zone per pitches seen inside Strike Zone	2002
Swing%	Swings per Pitches seen	2002
O-Contact%	Contacted Pitches Outside Strike Zone per Swings at Pitches Outside Strike Zone	2002
Z-Contact%	Contacted Pitches Inside Strike Zone per Swings at Pitches Inside Strike Zone	2002
Contact%	Contacted Pitches per Swings	2002
Pull%	Pulled Field Balls in Play per Total Balls in Play	2002
Cent%	Center Field Balls in Play per Total Balls in Play	2002
Oppo%	Opposite Field Balls in Play per Total Balls in Play	2002
Soft%	Soft Contact per Balls in Play	2002
Med%	Medium Contact per Balls in Play	2002
Hard%*	Hard Contact per Balls in Play	2002
EV	Average Exit Velocity	2015
LA	Launch Angle	2015
Barrel%	Barreled Balls per Balls in Play	2015
maxEV	Max Exit Velocity	2015
HardHit%*	Hard Hit Balls per Balls in Play	2015
wOBA	Weighted on Base Average	2015

**Note that HardHit% and Hard% are two different metrics attempting to measure the same thing. Hard% is provided by Baseball Info Solutions (BIS) and is calculated via video scouts using a proprietary algorithm. HardHit% is provided by Statcast which counts any ball hit over 95 MPH.*

Training Sets

Training Set	Description	Number Samples	K-Neighbors	Decision Tree Depth	PCA Dimensions
1-Year Basic	1 Year of data using features available in all years without imputation	6707	20	5	6
1-Year Advanced	1 Year of data using features available in 2002 and earlier without imputation	1459	15	5	10
1-Year All	1 Year of data using all features without imputation	180	10	5	10
2-Year All	2 Years of data using all features without imputation	59	5	3	4
1-Year DT Imputed	1 Year of data using all features imputed with Decision Trees	6707	5	5	7
2-Year DT Partial Imputed	2 Years of data using all features. Season X data imputed with Decision Trees. Season X-1 data not imputed	4750	10	5	5
2-Year DT Imputed	2 Years of data using all features imputed with Decision Trees	6707	10	5	5
3-Year DT Partial Imputed	3 Years of data using all features. Season X data imputed with Decision Trees. Season X-1, X-2 data not imputed	3465	5	7	5
3-Year DT Imputed	3 Years of data using all features imputed with Decision Trees	6707	10	7	10
1-Year LR Imputed	1 Year of data using all features imputed with Linear Regression	6707	10	5	5
2-Year LR Partial Imputed	2 Years of data using all features. Season X data imputed with Linear Regression. Season X-1 data not imputed	4750	20	7	8
2-Year LR Imputed	2 Years of data using all features imputed with Linear Regression	6707	20	5	9
3-Year LR Partial Imputed	3 Years of data using all features. Season X data imputed with Linear Regression. Season X-1, X-2 data not imputed	3465	15	5	13
3-Year LR Imputed	3 Years of data using all features imputed with Linear Regression	6707	10	5	10

Parameter Tuning Example

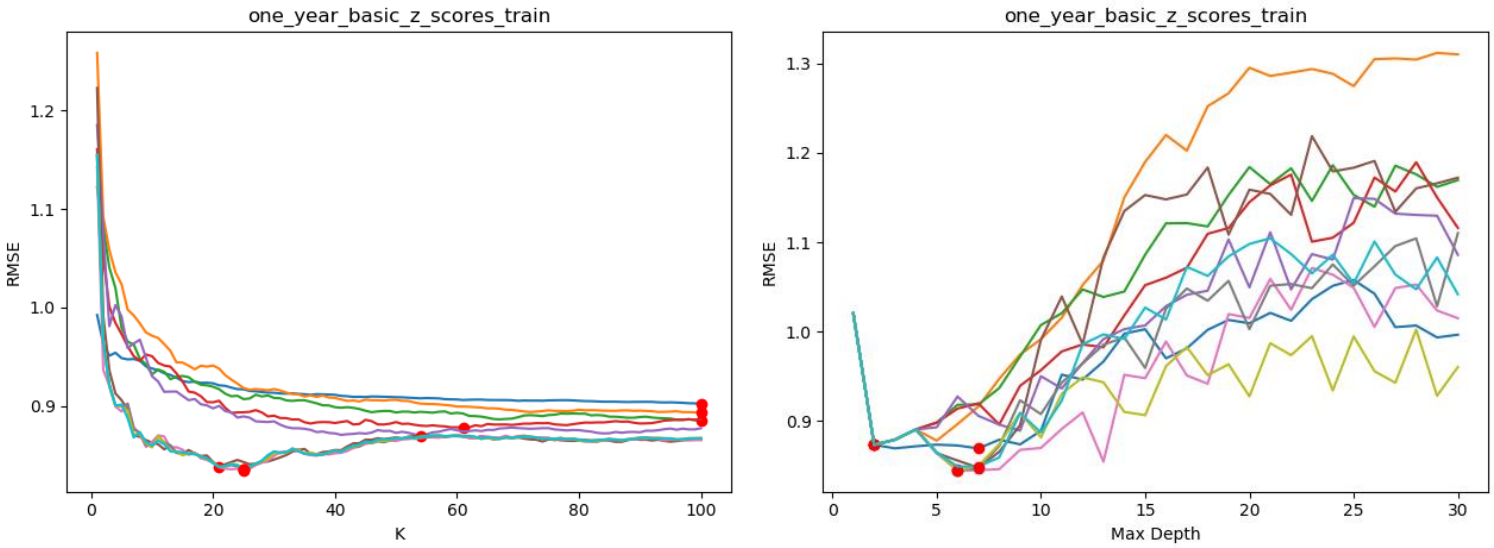


Figure 1 In the image on the left the x-axis is the parameter k used in K-Nearest Neighbors. On the right, the x-axis is the max depth of the Decision Trees. In both, the y-axis is the RMSE and each line represents a different number of dimensions after PCA. K-Nearest Neighbors is optimal around k=20 and Decision Tees around depth=5 where the lines appear to flatten before significant overfitting.

PCA Dimensions Example

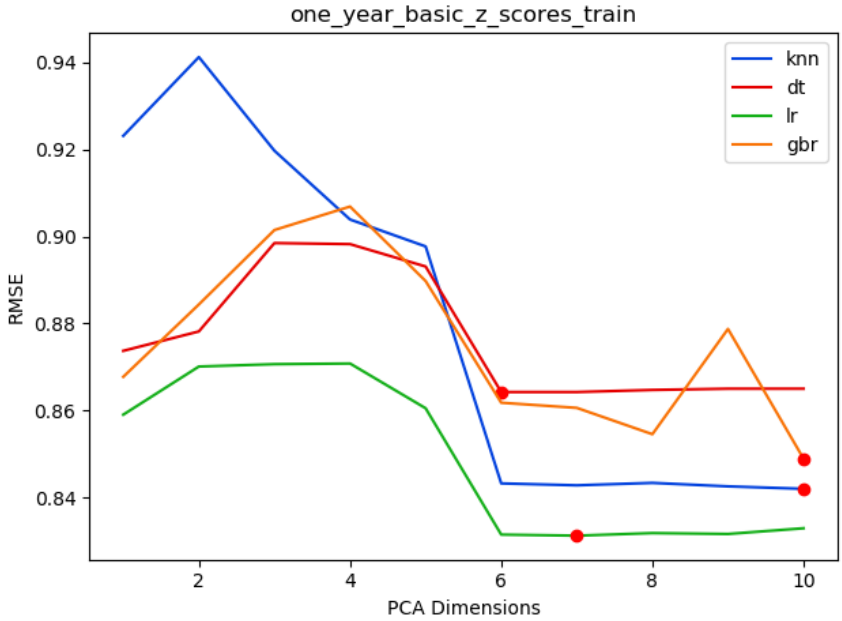


Figure 2 The x-axis is the number of dimensions used in PCA and the y-axis is the RMSE of each model. All four lines flatten around 6 dimensions so this was the chosen value for this training set.

Bibliography

Ishii, T. (2016). *Using Machine Learning Algorithms to Identify Undervalued Baseball Players*. Retrieved from http://cs229.stanford.edu/proj2016/report/Ishii-UsingMachineLearningAlgorithmsToIdentifyUndervaluedBaseballPlayers-report.pdf

Judge, J. (2020, July 7). *The Delta Method, Revisited: Rethinking Aging Curves*. Retrieved from Baseball Prospectus: https://www.baseballprospectus.com/news/article/59972/the-delta-method-revisited/

Lichtman, M. (2009, December 2009). *How do baseball players age? (Part 2)*. Retrieved from The Hardball Times: https://tht.fangraphs.com/how-do-baseball-players-age-part-2/

PECOTA. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/PECOTA

Szymborski, D. (2020, November 10). *The 2021 ZIPS Projections: An Introduction*. Retrieved from Fangraphs: https://blogs.fangraphs.com/the-2021-zips-projections-an-introduction/

Tango, T. (2004, March 10). *The 2004 Marcels*. Retrieved from TangoTiger: http://www.tangotiger.net/archives/stud0346.shtml

Tango, T. (2012). *Marcel*. Retrieved from TangoTiger: http://www.tangotiger.net/marcel/

Tango, T. M., Lichtman, M. G., & Dolphin, A. E. (2006). *The Book: Playing the Percentages in Baseball.* Newark: TMA Press.