REGRESSION ANALYSIS TO
PREDICT WINS ABOVE REPLACEMENT

Assessing model accuracy using offensive baseball data to project wins above replacement

By
Zach Tallevast

Statistics

Stat 4002
Dr. Erin Schliep

February 2018

Table of Contents:

Introduction:

In sabermetrics (the study of baseball statistics), analysts often use a comprehensive baseball statistic known as WAR or “Wins Above Replacement”. This statistic was designed to reflect accurately how a given player contributes to their team in an individual season.

WAR is defined as “The number of statistical wins the player is responsible for above a replacement player”. Theoretically a replacement player is a player at league average but for this analysis, we will use positive and negative war for “contribution” or “no contribution”.

This WAR statistic is built by FANGRAPHS which is a baseball statistics company who built the statistic internally to provide a holistic metric interpreted for player value to allow comparisons across a team, the league, years, and eras. Over time, many baseball statisticians have created formulas to estimate a WAR value by player, but none have reciprocated the metric perfectly.

In this analysis, I will attempt to build a classification model to accurately estimate WAR as positive or negative.

Question: Is there a relationship between these observed offensive statistics and WAR?

Data Selection:

The data selected is from Fangraphs offensive tables. The data is a subset containing 2016 and 2017 offensive data with 53 variables and 290 observations.

51 Variables are Numerical, and 3 Variables are categorical (Team and Player Name and Classification)

For Model 1: I completed a multiple linear regression model to compute WAR numerically.

For Models 2-6: I completed Classification models to predict the variable Classification.

Classification:

Positive: For Players with Positive WAR.

Negative: For Players with Negative WAR.

Model 1: Multiple Linear Regression

$WAR = B_0 + B_1X_1 + B_2X_2 + \ldots\ldots\ldots + B_{51}X_{51}$

Where X_n corresponds to our numerical predictors.

Initial Model:

WAR ~ G + PA + HR + R + RBI + SB + BB. + K. + ISO + BABIP + AVG + OBP + SLG + wOBA + wRC. + BsR + Off + Def + BB.K + Spd + UBR + wGDP + wSB + wRC + wRAA + GB.FB + LD. + GB. + FB. + IFFB. + HR.FB + IFH + IFH. + BUH + BUH. + Pull. + Cent. + Oppo. + Soft. + Med. + Hard. + O.Swing. + Z.Swing. + Swing. + O.Contact. + Z.Contact. + Contact. + Zone. + F.Strike. +SwStr.

Stepwise selection determined that the best model uses the significant predictors of:

For a Final model:

```
Call:
formula = WAR ~ PA + SB + BB. + ISO + BABIP + OBP + wOBA + wRC. + BsR + Off + Def
+ UBR + wGDP + wRAA + IFFB. + Oppo. + O.Swing. + Swing. + Z.Contact.
Residuals:
     Min        1Q    Median       3Q       Max
-0.190485 -0.050327  0.002069  0.052428  0.182459
```

Coefficients Table	Estimate	Std. Error	t-value	Pr(>|t|)
Intercept	2.337e+00	5.373e-01	4.349	1.94e-05***
PA	3.555e-03	9.335e-05	38.083	<2e-16***
SB	-1.222e-03	7.925e-04	-1.542	0.124142
BB%	1.128e+00	5.663e-01	1.992	0.047375*
ISO	-2.611e+00	6.496e-01	-4.019	7.59e-05***
BABIP	-7.232e-01	3.269e-01	-2.212	0.027787 *
OBP	-4.934e+00	1.461e+00	-3.377	0.000841 ***
wOBA	-5.181e+00	3.660e+00	-1.416	0.158057
wRC+	1.942e-02	4.387e-03	4.426	1.40e-05***
BsR	2.789e-02	8.156e-03	3.419	0.000725 ***
Off	8.248e-02	5.999e-03	13.749	<2e-16 ***
Def	1.009e-01	5.145e-04	196.063	<2e-16 ***
UBR	-8.500e-03	5.586e-03	-1.522	0.129251
wGDP	-1.524e-02	6.368e-03	-2.394	0.017357 *
wRAA	1.622e-02	5.328e-03	3.045	0.002555 **
IFFB%	1.611e-01	1.077e-01	1.496	0.135820
Oppo%	-3.008e-01	1.676e-01	-1.795	0.073793
O.Swing%	6.374e-01	2.092e-01	3.047	0.002539 **
Swing%	-6.009e-01	2.323e-01	-2.587	0.010206 *
Z. Contact%	-4.985e-01	1.679e-01	-2.969	0.003254 **

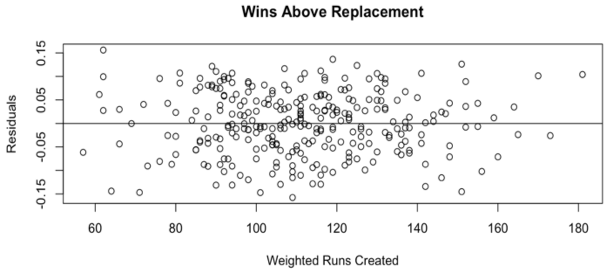
Significance Codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

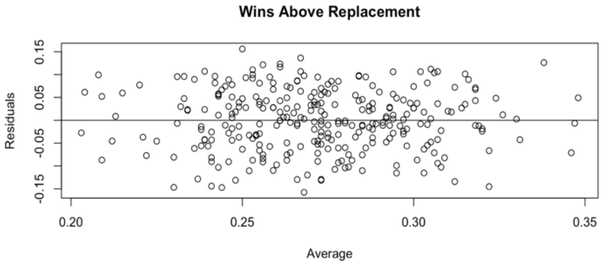
Residual standard error: 0.07067 on 270 degrees of freedom

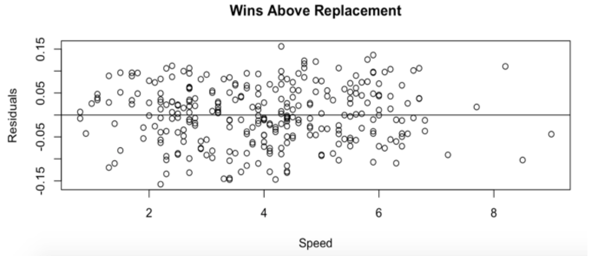
Multiple R-squared: 0.9988, Adjusted R-squared: 0.9987

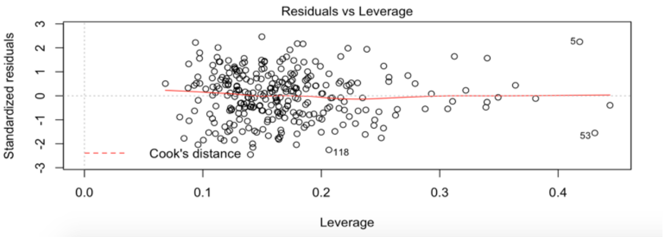
F-statistic: 1.204e+04 on 19 and 270 DF, p-value: < 2.2e-16

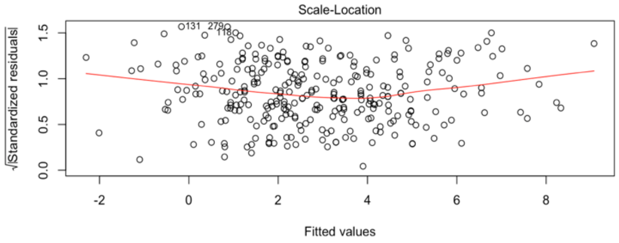
Residual Plots:

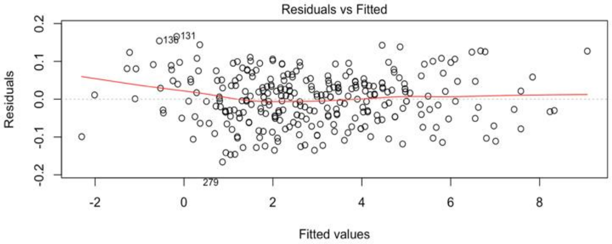










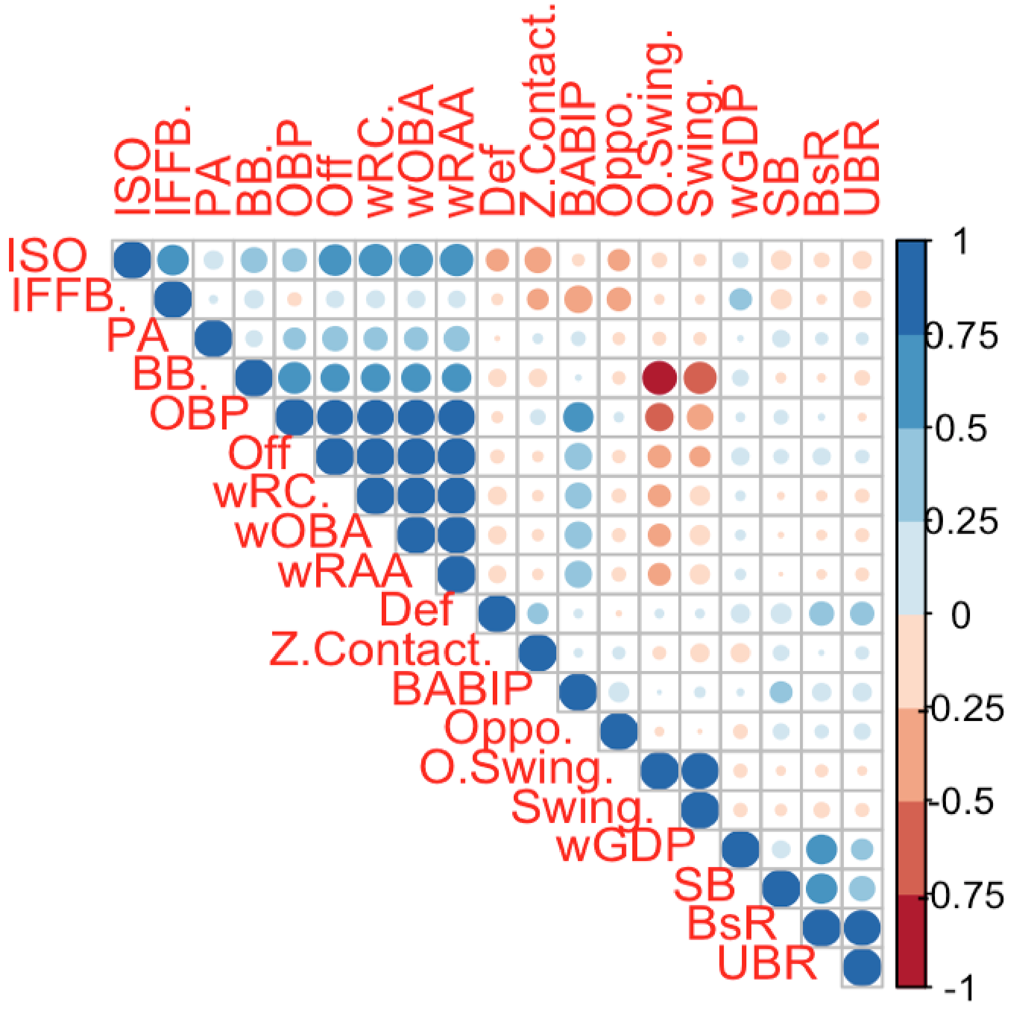


The residual plots for speed, weighted runs created, and leverage suggest that a linear regression model is appropriate for this data. The residual vs fitted plot shows that this model is nearly linear, and this is a significant model. The scale-location plot shows that the model lacks some homoscedasticity (or the assumption of equal variance is not perfect). The Residuals vs. Leverage plot shows that no outliers are influencing the models R-Square value.

Test for Multicollinearity:

```
Overall Multicollinearity Diagnostics
                         MC Results detection
Determinant |X'X|:       0.000000e+00         1
Farrar Chi-Square:                Inf         1
Red Indicator:          3.210000e-01         0
Sum of Lambda Inverse: -5.118666e+13         0
Theil's Method:         -2.496400e+00         0
Condition Number:                 NaN        NA
1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test
===================================
```

Correlation Matrix



As seen in the correlation matrix, there exists high correlation between OBP, OFF, wRC, wOBA, and wRAA. This suggests this data has multicollinearity.

Model 2: (Logistic Regression Analysis)

Due to bias, I had to throw out OFF, DEF, IFFB%, Swing%, and Z.Contact. These predictors did not have enough variability to use in a classification analysis.
Training Set has 182 Rows and my Test Set has 108 Rows.

Classification = PA+SB+F.Strike+ISO+AVG+OBP+wOBA+wRC+BsR+BB/K%+UBR+wGDP+wRAA+GB%+IFH%+BUH+O.Swing

The smallest p-values for LR are Plate Appearances (PA), Average (AVG), and Weighted Runs Created (wRC+). Therefore, there is association between Classification and PA, AVG, and wRC+.
My Test AIC = 84.691.

Actual Test Results	Predicted Results		
	glm.pred	Negative	Positive
	Negative	11	1
	Positive	2	168

Accuracy: 98.35%
Misclassification Rate: 1.65%
Sensitivity: 98.88%
False Positive Rate: 8.33%
Logistic Regression accurately predicted a players WAR classification with a threshold of 50%, 98.35% of the time.

To check if Binary Logistic Regression is statistically significant, you can test using a goodness of fit test. I used the Hosmer-Lemeshow test.

Hosmer-Lemeshow H statistic
X-squared = 182, df = 8, p-value < 2.2e-16

This P-Value is less than alpha = 0.05 so I can conclude that Logistic Regression does not fit this dataset.

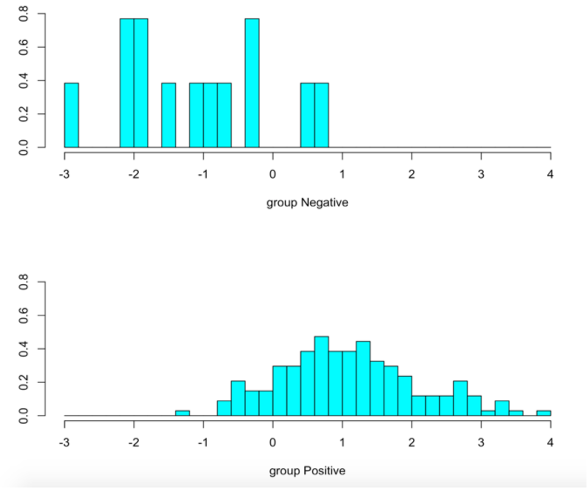
Model 3: (Linear Discriminant Analysis)

$$\underline{\boldsymbol{pi(1) = 0.0714\ and\ pi(2) = 92.857}}$$

7% of the training observations correspond to negative war classifications.

Actual Test Results	Predicted Results		
	lda.class	Negative	Positive
	Negative	6	1
	Positive	7	168

Accuracy: 95.60%
Misclassification Rate: =4.40%
Sensitivity: 96.0%
False Positive Rate: 14.29%
Linear Discriminant Analysis accurately predicted a players WAR classification 95.60% of the time.



The negative and positive classes here are well distinguished. The positive group is centered around 1 and has consistent spread. The negative class does not have a center or equal spread which suggests that the class separation was not a good predictor for WAR.

Model 5: (Quadratic Discriminant Analysis)

Could not compile due to number of predictors.

Model 6: (K-Nearest Neighbors Approach)

For k=3,

The Confusion matrix is below:

	Predicted Results		
Actual Test Results	knn.pred	Negative	Positive
	Negative	5	1
	Positive	8	168

Accuracy: 95.05%

Misclassification Rate: 4.95 %

Sensitivity: 95.45%

False Positive Rate: 16.67%

K-Nearest Neighbors for k=3 accurately predicted a players WAR classification 95.05% of the time.

Summary:

For Binary Logistic Regression Analysis, the model resulted in the largest model accuracy. Although our model did not achieve high p-valued predictors, this model resulted in the least number of false positives and the highest level of sensitivity for this data. While this model looked to be significant, a goodness of fit test deemed this fit as insignificant.

For Linear Discriminant Analysis, the model resulted in the second lowest model accuracy. This is due to the Y-Classes are not well separated in this dataset. Since there is 5 times as many positive WAR players as negative WAR players, this approach was unreliable. The assumption of uniform variance was slightly off so LDA has some bias. This is due to with WAR, players can achieve the same x number of WAR points in different number of games played. So, the predictors used to calculate WAR, are far from linear.

K-Nearest Neighbors approach resulted in the lowest model accuracy. With this large number of predictors, KNN fails to record reliable subgroup MSE Values at each k. Although the data is likely non-parametric as we see with the significance of the other models, KNN suffers tremendously with this number of predictors. Linear regression offers increased interpretability. KNN also resulted in the largest false positive rate.

Classification Model Type	**Accuracy**	**Significance**
Binary Logistic Regression	98.35%	Does not fit data
Linear Discriminant Analysis	95.60%	Check with Cross Validation
k-Nearest Neighbors	95.05%	Does not fit data

In conclusion, to better this classification modeling approach, I would gather larger data and separate the data into more classes with equal sizes and utilize a linear discriminant analysis approach to achieve more perfect model accuracy.

Additional Analysis:

For further analysis, instead of WAR as a binary response of 1: Player has worth and 2: Player has no worth, another approach would be to break WAR into categories such as (Elite, Average, Below Average) to classify players into subgroups of equal sizes. This would improve the Linear Discriminant Analysis model as there is not such a drastic difference in the sample sizes of the binary response.

For better data, it would be beneficial to look at WAR across a decade, instead of across two seasons. Since k-Nearest Neighbors did not produce high accuracy with the small sample, I can conclude that KNN is not an approach that can properly model WAR because as sample size increases, KNN becomes less predictive.