MONEYBALL OLS REGRESSION PROJECT

Christopher Jose - Predict 411 – Sec 60
Kaggle Name: Christopher Jose

Introduction

Using statistics is an absolutely essential tool in determining the success of a baseball team. Long gone are the days of old where teams mainly relied on the wits of their recruiters and staff members to determine what is in the best interest of the team. I analyze baseball statistics, standardized to match a 162 game season, to construct models that predict the number of wins for a team over the course of a season. All of the models I consider show an incredible level of predictability that baseball statistics offer in determining the number of games won.

1. Exploratory Data Analysis

Dataset characteristics

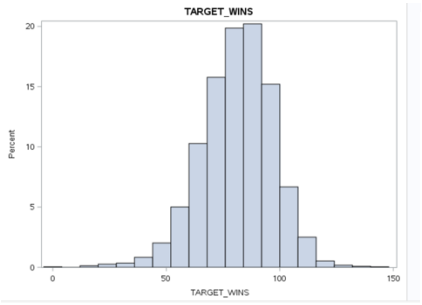
The dataset contains 2276 observations and 16 variables. Each observation contains statistics on the performance of a professional baseball team from the years 1871 to 2006 inclusive. Each statistic is adjusted to match the performance of a 162 game season. All variables are numeric. 15 of the variables will be used in linear regression to predict the 16th variable, TARGET_WINS, the number of wins for the team. A data dictionary of the variables is as follows:

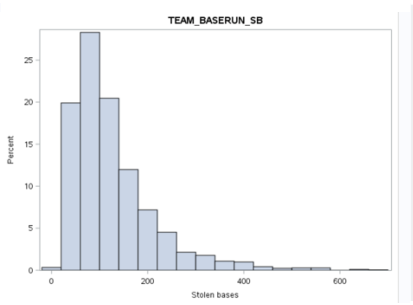
Alphabetic List of Variables and Attributes

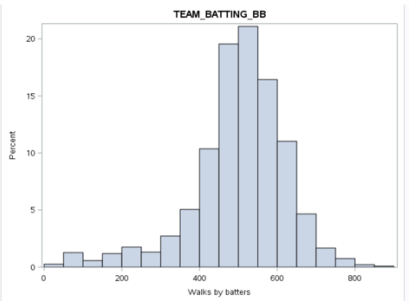
#	Variable	Type	Len	Label
1	INDEX	Num	8	
2	TARGET_WINS	Num	8	
10	TEAM_BASERUN_CS	Num	8	Caught stealing
9	TEAM_BASERUN_SB	Num	8	Stolen bases
4	TEAM_BATTING_2B	Num	8	Doubles by batters
5	TEAM_BATTING_3B	Num	8	Triples by batters
7	TEAM_BATTING_BB	Num	8	Walks by batters
3	TEAM_BATTING_H	Num	8	Base Hits by batters
11	TEAM_BATTING_HBP	Num	8	Batters hit by pitch
6	TEAM_BATTING_HR	Num	8	Homeruns by batters
8	TEAM_BATTING_SO	Num	8	Strikeouts by batters
17	TEAM_FIELDING_DP	Num	8	Double Plays
16	TEAM_FIELDING_E	Num	8	Errors
14	TEAM_PITCHING_BB	Num	8	Walks allowed
12	TEAM_PITCHING_H	Num	8	Hits allowed
13	TEAM_PITCHING_HR	Num	8	Homeruns allowed
15	TEAM_PITCHING_SO	Num	8	Strikeouts by pitchers

To get a sense of the dependent variable's data, TARGET_WINS, consider the following: it has a minimum of 0, mean of 81, maximum of 146 (the total possible value is 162—the total number

of games per season), and appears to be normally distributed, as shown in leftmost histogram below. In comparison, the predictor variables do not often appear normally distributed. Four predictors clearly appear to have textbook right-skewed distributions (middle plot), and one predictor is moderately left-skewed (right plot). This is a tipoff that the dependent variable is fairly tame, though the predictors will have outliers, which I will better confirm with boxplots.







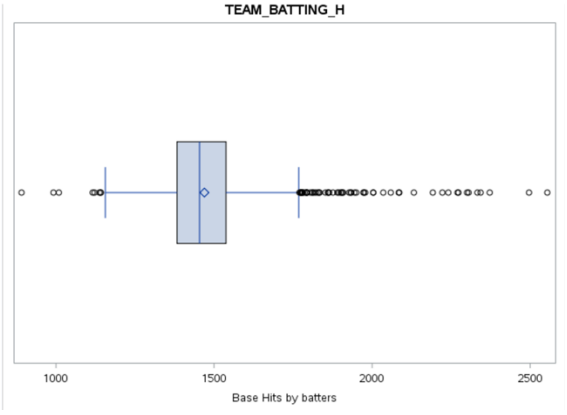
Identifying Outliers by Inspecting Boxplots

I check for outliers by examining box plots for each variable. I am using Tukey's method to classify points as outliers, meaning any value beyond the interquartile range (IQR) by an amount of (1.5 x interquartile distance) is identified as an outlier. In other words, any value satisfying the following formula is identified as an outlier:

X< (25^{th} percentile) - 1.5 (interquartile distance) aka X< Q1 - 1.5(IQR)
X> (75^{th} percentile) + 1.5 (interquartile distance) aka X> Q3 + 1.5(IQR)

On the boxplots, all points identified with markers that are outside the box and its whiskers are outliers as defined by Tukey's method. Over half the variables appear to have a sizable number of outliers that sometimes extend quite far from the whiskers of the boxplot.

The boxplot for TEAM_BATTING_H shown below, for example. All dots are outliers.



Identifying missing values

I observe missing values for six variables. Four of the missing variables have less than 10% of their values missing. As for the other two missing variables: 92% of TEAM_BATTING_HBP's values are missing, 34% of TEAM_BASERUN_CS's values are missing. The number of missing values for each variable is shown in the **N Miss** column of **The MEANS Procedure** table.

Trimmed Variable Information – Original missing values are preserved

The MEANS Procedure

Variable	Label	N	N Miss	Mean	Median	Minimum	Maximum
INDEX		2276	0	1268.46	1270.50	1.0000000	2535.00
TARGET_WINS		2276	0	80.7908612	82.0000000	0	146.0000000
TEAM_BATTING_H	Base Hits by batters	2276	0	1469.27	1454.00	891.0000000	2554.00
TEAM_BATTING_2B	Doubles by batters	2276	0	241.2469244	238.0000000	69.0000000	458.0000000
TEAM_BATTING_3B	Triples by batters	2276	0	55.2500000	47.0000000	0	223.0000000
TEAM_BATTING_HR	Homeruns by batters	2276	0	99.6120387	102.0000000	0	264.0000000
TEAM_BATTING_BB	Walks by batters	2276	0	501.5588752	512.0000000	0	878.0000000
TEAM_BATTING_SO	Strikeouts by batters	2174	102	735.6053358	750.0000000	0	1399.00
TEAM_BASERUN_SB	Stolen bases	2145	131	124.7617716	101.0000000	0	697.0000000
TEAM_BASERUN_CS	Caught stealing	1504	772	52.8038564	49.0000000	0	201.0000000
TEAM_BATTING_HBP	Batters hit by pitch	191	2085	59.3560209	58.0000000	29.0000000	95.0000000
TEAM_PITCHING_H	Hits allowed	2276	0	1779.21	1518.00	1137.00	30132.00
TEAM_PITCHING_HR	Homeruns allowed	2276	0	105.6985940	107.0000000	0	343.0000000
TEAM_PITCHING_BB	Walks allowed	2276	0	553.0079086	536.5000000	0	3645.00
TEAM_PITCHING_SO	Strikeouts by pitchers	2174	102	817.7304508	813.5000000	0	19278.00
TEAM_FIELDING_E	Errors	2276	0	246.4806678	159.0000000	65.0000000	1898.00
TEAM_FIELDING_DP	Double Plays	1990	286	146.3879397	149.0000000	52.0000000	228.0000000

Variable Correlations

I examine Pearson correlation coefficients between predictors and the dependent variable, TARGET_WINS. One variable, TEAM_BATTING_H, has a moderate correlation with the predictor (.38877). Nine variables have weak correlations with the predictor (|.1| to |.3|), 2 of which are negative correlations (the remaining are positive). The remaining five variables have very weak correlations (<|.1|), where 3 are negative. These correlations suggest that a linear regression model using these predictors might yield statistically significant predictor coefficients.

Also, TEAM_FIELDING_DP, the number of double plays, is negatively correlated with TARGET_WINS by an amount of -0.03485. While this may be counter-intuitive, this is what the data is saying. Thus, we should expect this predictor's estimated coefficient to also be negative.

In examining correlations between predictors, I observe:
6 variables are strongly correlated (|.5| to |1|) with 3 or more other predictors
1 variable is strongly correlated with 2 other predictors
4 variables are strongly correlated with 1 other predictor
4 variables are not strongly correlated with any other predictors

TEAM_FIELDING_E is strongly correlated with the highest number of predictors (6). Only two correlations are above |.7| and both are for TEAM_BATTING_HR, which has a .96937 correlation with TEAM_PITCHING_HR and a .72707 correlation with TEAM_BATTING_SO. These correlations among the predictors suggests that the model might have multicollinearity problems, which we will more formally examine later by inspecting variance inflation factors (VIFs).

Multicollinearity can cause large standard errors for predictor coefficients, and therefore insignificant coefficients. This would mean we won't have statistical evidence that these coefficients are greater than zero when they might actually be, meaning we won't have a good estimate of what these coefficients actually are. Thus, we won't be precisely estimating the actual relationship between predictors and target wins, meaning the predicted change in target wins from changing predictors that are insignificant as a result of multicollinearity, holding all other variables constant. However, the good thing is that the predictability of the model will be unaffected and we can still make good predictions (even if some of our estimated coefficients are off from their actual values).

2. Data Preparation

Trimming the data to remove outliers

I transform the data by trimming (limiting) values of each variable to be within (1.5 x interquartile distance) of each variable's interquartile range (IQR). I've decided to trim the data to limit the effect that these outliers will have on the regression line, which will likely be influenced, and potentially in a negative way such that its predictions are worsened (less reflective of the likely value, and more reflective of rare, extreme values). Even though the dependent variable's data appears to have less of an outlier problem and is more tamed, as shown by its histogram, I have trimmed its data too so that it is on the same basis as the independent variables (in that its data is also trimmed).

Imputing missing values with the average value

For each variable with missing values, I impute the mean of its trimmed data. Trimming the variables does result in the imputation of missing values, which sets them equal to the threshold used in the trimming process. I want to do mean imputation, so I change these values back to missing again and then fill them in with each variable's trimmed mean.

I trim outliers before imputing data because if mean imputation were to come first, then for certain variables with many missing values, the data transformation would become distorted. For instance, mean imputation before trimming for variable IMP_TEAM_BATTING_HBP results in Q3=Q1=mean, resulting in IQR = 0. If the variable is then trimmed, all values are reduced to or increased to be equal to the mean, which takes away the information that this variable contains.

Examining correlations after having transformed the data

Transformed variable correlations still maintain their original level of having either very weak, weak, or moderate correlations with the predictor. TEAM_BATTING_H's still has a moderate correlation (.38079, which is down from .38877). Nine variables still have moderate correlations (between |.1| and |.3|). So I expect the transformed variables to still be predictive of TARGET_WINS.

3. The Models

I build models using the following techniques:

Stepwise Regression
Maximum Adjusted R-Squared Improvement Regression Technique
Random Forests

Models will be built by splitting the data into train and test splits. 70% of the data will be randomly selected for training the models, while the remaining 30% will be used to evaluate the performance of the models. However, for Random Forests, I construct one model from training data, and another model from all of the data.

All variables starting with the letters "imp_" indicate that the variable had missing values, which I filled with that variable's trimmed mean. In using any of the models for future data, you would just use the original variable in that variable's place (unless it has missing values).

The key metric that I will use to evaluate models is the root mean squared error (rmse). I will consider the adjusted r-squared as being of secondary importance. The rmse will be calculated on the test data (except for one of the Random Forest models), while the adjusted r-squared is calculated on the training data. Rmse thus answers the question: what is the average error in predicting the test data using this model? Adjusted r-squared answers the question: how well does our model, built from the training data, explain the variation in the target wins values in the training data?

Stepwise Regression

Trimmed, Mean Imputed Data

This technique consists of multiple steps, where at each step:

1) A predictor is entered into the model if it is the best predictor of the dependent variable, relative to other predictors.
2) A predictor is removed from the model if it is not a good predictor of the dependent variable.

Whether or not a predictor is evaluated as deserving to be added or removed from the model is based on a numerical measurement of how sure we are that the predictor is predictive of the dependent variable. Otherwise known as the p-value of the predictor, it is the likelihood that the predictor's estimated coefficient differs from zero.

The model consists of 7 predictors and is as follows: predicted wins =
18.47341 + .04267(imp_team_baserun_sb) + .13036(team_batting_3b)
+.02071(team_batting_bb) + .04066(team_batting_h) - .10881(imp_team_fielding_dp)
-.03973(team_fielding_e) + .04169(team_pitching_hr)

All estimated coefficients add a significant amount of predictability to the number of wins and are statistically significant at the .01% level, assuming regression assumptions hold relatively well. After inspecting certain diagnostics, I conclude that they do hold relatively well. Thus, all predictors definitely appear to have a relationship with and are predictive of target wins. This does not necessarily mean that the model will actually do a good job predicting target wins though. A measurement of the predictability of the model is the adjusted r-squared, which is .2806, and indicates that the model explains 28.06% of the variation in target wins. The model will thus offer predictability, but imprecise predictability at times. The other measure of predictability is the root mean squared error (rmse), the typical amount by which a predicted value deviates from the actual value. I use the model to predict test data values on the test split, which yields an rmse of 13.95.

In examining regression coefficients to see whether they make sense, I observe the following: Regression coefficients make sense, except for imp_team_fielding_dp and team_pitching_hr, imp_team_fielding_dp indicates the number of double plays (making two outs in one play), and has a negative coefficient of -.10881. This does not make intuitive sense, however, as I previously mentioned, its correlation coefficient with target wins is negative. This is what the data is saying, and while it does not seem to make sense, the variable is statistically significant at the .001 level, so I leave it in. Variable team_pitching_hr has the same story. It represents the number of allowed homeruns, which one would think would have a negative impact on number of wins. However, its correlation with wins is positive and its estimated correlation coefficient is positive. Since the coefficient is significant and predictive, so I will leave it in.

In assessing regression assumptions, I inspect the residual plot, normal q-q plot, and Cook's D plot. Assumptions appear to hold: residuals are relatively homoscedastic, normally distributed, are uncorrelated, and there are no influential outliers.

All variance inflation factors (VIFs) are under 4, suggesting that multicollinearity is not an issue.

Maximum Adjusted R-Squared Improvement Regression Technique
Trimmed, Mean Imputed Data

This technique consists of multiple steps that add and remove variables only if they increase the adjusted R-squared (adjr2). Specifically, the technique starts with a variable that leads to the highest adjr2, creating a one-variable model. Then it adds the variable that leads to the largest increase in adjr2, creating a two-variable model. Then it considers replacing either of these two variables with a variable that would lead to a higher adjr2, which optimizes the two-variable model. Then it adds a third variable which leads to the largest increase in adjr2, creating a three-variable model. This process continues until the adjr2 is no longer increased.

The model consists of 12 predictors and is as follows: predicted wins =
33.23446 - .07039(imp_team_baserun_cs) + .06586(imp_team_baserun_sb)
+ .13678(team_batting_3b) + .04537(team_batting_bb) + .03599(team_batting_h)
+ .09190(team_batting_hr) - .03788(imp_team_batting_so) - .09914 (imp_team_fielding_dp)
- .06055(team_fielding_e) - .02342(team_pitching_bb) - .03533(team_pitching_hr)
+ .02828 (imp_team_pitching_so)

10 variables add a significant amount of predictability to the model in that they are significant at the 1% level. Variable team_batting_hr has a p-value of .018, meaning it is also a fairly significant predictor. Variable team_pitching_hr does not have a coefficient that differs enough from zero to suggest that this predictor is predictive of target wins.

In examining regression coefficients that are counterintuitive:
Similar to the Stepwise Regression model, imp_team_fielding_dp also has a negative coefficient, which is because it is negatively correlated with target wins. It will be kept in the model (also because it contributes towards maximizing the adjr2. Other coefficients make sense: for instance, team_pitching_hr, the number of allowed homeruns, has a negative coefficient, which is what we would expect. As allowed homeruns goes up, the other team does better, and predicted number of wins goes down.

The adjusted r-squared of this model only increases slightly to 29.50% (a smidge up from 28.06%). The rmse on the test data is 13.75 (down from stepwise regression's 13.95).

Regression assumptions appear to hold reasonably well, based on inspecting the residual plot, normal q-q plot, and Cook's D plot.

VIFs for several predictors are above 10, suggesting that these predictors are highly correlated with other predictors. As a rule of thumb, VIFs greater than 5 indicate multicollinearity problems. Multicollinearity can distort coefficient estimates by resulting in bigger estimated coefficient standard errors, and thus less significant predictors, meaning we won't be sure if the effected predictors are actually predictive of the dependent variable. The two predictors with the highest VIFs also have the highest p-values (team_batting_hr and team_pitching_hr).

Maximum Adjusted R-Squared Improvement Regression Technique with variable reduction
Trimmed, Mean Imputed Data

One variable removed
I implement the same algorithm before, but with team_batting_hr removed, which had the highest VIF. The resultant model has the same predictors (excluding the dropped predictor), all statistically significant at the 1% level. There are 11 predictors in this model. The adjusted r-squared decreases slightly to 29.29% (from 29.50%). The rmse on the test data increases very slightly to 13.93 (from 13.75). Regression assumptions still hold.

Predictor coefficients change very little, except for team_pitching_hr, whose coefficient changes from negative to positive. Interestingly, while this predictor's coefficient no longer makes intuitive sense, it is now statistically significant (it used to be insignificant, with a p-value of .2971), and its VIF has decreased from 41 to 3.9, so I leave it in the model.

Three variables removed
VIFs for three predictors are still above 10 in the above model, which is troubling, so I run the same algorithm with the predictors of the 2 largest VIFs of the above model removed as well (imp_team_batting_so and imp_team_pitching_so). These variables have VIFs of 12.5 and 19.13, respectively.

All predictors stay in the model, except for those removed, and one new predictor is added (team_pitching_h). Thus there are 10 predictors in this model. Predictor coefficients change very little. The coefficient on team_pitching_h, meaning hits allowed, is positive, when we would think it should be negative. However, this predictor is significant at the 1% level (though it's VIF is above 5, which is troubling). All predictors are significant at the 1% level, except for team_pitching_bb (whose p-value went up), though this predictor is significant at the 5% level, and is thus still very predictive. Regression assumptions hold. The adjusted r-squared drops to 28.71%. The rmse on the test data decreases to 13.49, making it the best rmse thus far.

Four variables removed
VIFs are still above 5 for several variables in the above model, so I run the algorithm again by excluding team_batting_bb, which has the highest VIF in the above model (8.91).

This model has 10 predictors because a new predictor is added (team_batting_2b), though this predictor is very insignificant, and its coefficient is counterintuitive. Another variable, team_pitching_h, is very insignificant as well (when it used to be significant). All other variables are significant at the 1% level and coefficients change very little. Regression assumptions hold. The adjusted r-squared drops to 27.90%. The rmse on the test data drops to 12.48, making it the best rmse thus far.

Five variables removed

Only one VIF in the above model remains above 5 (team_fielding_e), so I remove this variable from the algorithm and observe the results. This model only has 7 predictors, while the previous model had 10, since it also did not include team_batting_2b and imp_team_baserun_cs. All predictors are significant at the 1% level and have VIFs less than 5. Regression assumptions hold. All coefficients change little, except for team_pitching_h, whose coefficient goes from positive to negative, and thus makes more sense and is now significant. The adjusted r-squared drops down to 24.73% (from 27.90%). The rmse on the test data goes up to 13.46.

Random Forests

This technique consists of constructing a pre-specified number of decision trees, each of which predicts the dependent variable. The predictions from all of the trees are then averaged to produce the actual prediction. In this case, the decision trees are regression trees because the dependent variable is numeric rather than categorical. Each tree is different because each is constructed using two randomization steps called tree bagging and feature bagging. The tree bragging randomization step causes each tree to be constructed from randomly selected observations from the training data (though each tree selects the same number of observations to be built from). The feature bagging randomization step causes a random sample of variables/features to be considered at each split in each tree (though this random sample has the same predetermined size across all splits and trees). This is why this method is called "Random Forests", since a random forest consists of a bunch of trees that are constructed under two layers of randomization.

I choose to construct 300 regression trees, which is typically regarded as a reasonable number of trees. More trees usually result in a better result, though typically after a couple hundred trees (or perhaps less), there is diminishing returns, while the algorithm computation time increases. I keep the size of the random subset of features at sqrt(number of features) = sqrt(15)=4, which is a recommended rule of thumb and is the default value in sas. I leave all other parameters at their default values.

The rmse on the test data is 12.27, the lowest of all methods. The power of Random Forests is thus very evident. I did little tuning to the model, and yet it yielded the lowest rmse of all methods.

I also construct a random forest model on ALL of the trimmed, mean imputed data (train and test), also using 300 trees and the default parameters. The resulting rmse on the test data is 4.86. We would expect this model to be accurate when predicting the test data, considering it includes the test data in making it. It is perhaps unfair to compare 4.86 to 12.27, since they are apples and oranges, though potentially using the random forest model from ALL of the data is viable, however.

4. Selecting a Model

Models will be mainly compared and ranked by their root mean square error (rmse) on the test data. Then, of secondary importance, is maximizing the adjusted r-squared. We care about model predictability, which both of these metrics assess. We want to minimize rmse on the test data and maximize adjusted r-squared on the training data. I will also consider model intuitiveness, regarding whether variable coefficients make sense and are significant.

Model	**RMSE**	**AdjR2**
Random Forests*	4.86	
Random Forests**	12.27	NA
Max Adj. RSQ regression – 4 vars removed	12.48	.2790
Max Adj. RSQ regression – 5 vars removed	13.46	.2473
Max Adj. RSQ regression – 3 vars removed	13.49	.2871
Max Adj. RSQ regression	13.75	.2950
Max Adj. RSQ regression – 1 var removed	13.93	.2971
Stepwise regression	13.95	.2806

*rmse on test data, but constructed from all trimmed, mean imputed data
**rmse on test data, and constructed from training trimmed, mean imputed data

I will select Random Forests (RF) as the optimal model in predicting the number of wins, since RF built using the training data offers a fairly lower rmse than the regression models. While RFs are not as intuitive, I am prioritizing rmse. The regression models that have comparable rmses suffer from having lower adjusted r-squareds, making them less appealing. Also, I've decided to use the RF model based on all of the data, so that I do not lose predictive power as a result of not including 30% of the observations that are in the test split.

Again, 4.86 is not that comparable with the other rmses', since the model was built using the test data. If we build the model to include the test data, then in predicting the test data's target wins, we will, most likely, do a much better job at it.

Conclusion

All regression models have statistically significant predictors that are clearly predictive of the number of wins, and offer value in determining how successful a team will be. The model I select, Random Forests, offers the lowest rmse in relation to the regression models. However, I decide to train the Random Forests model on all of the trimmed, mean imputed data, so as to capture the full information of the data and potentially maximize the predictive power of the model.

Regarding future modelling improvements, I suggest implementing additional models, like Gradient Boosted Regression Trees (GBT). Also, I suggest tuning the GBT and Random Forest parameters using K-fold cross validation and grid search, which would undoubtedly lower the Random Forest's rmse even more and result in a low rmse value for the GBT technique.