Moneyball Assignment

By

Robert Torp

For

Predict 411 Section 55

Bonus Bingo (work follows the CODE section):

(20 Points) Use decision tree software such as Angoss for variable selection.

- I used a decision tree in Angoss KnowledgeStudio to select variables for one of the three models evaluated screenshots provided.

(10 Points) Use SAS Macros or use, in my opinion, good programming technique

- Several SAS Macros were used for in my code.

(10 Points) Hand in your SCORED FILE as a SAS DATA SET and save me to trouble of converting it.

- I included code that created a SAS DATA SET for my SCORED FILE.

(40 Points) - Total Bonus Bingo point attempted.

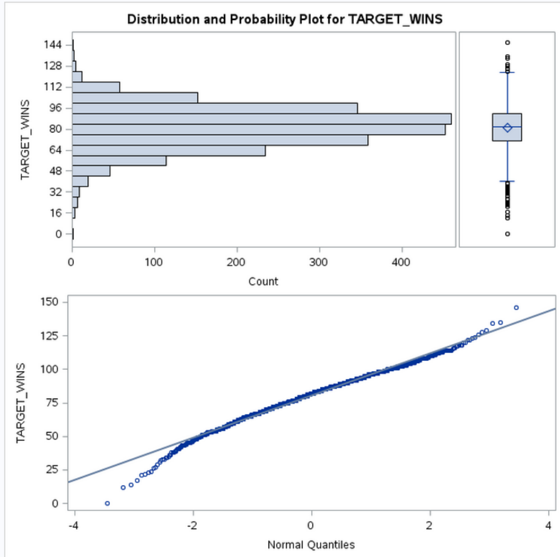
All Bonus Bingo work is summarized at the very end of the assignment after the CODE section.

Introduction

The purpose of this assignment is to analyze historical baseball data from 1871 to 2006 and determine if it is possible to develop a predictive model that can accurately predict how many wins a team will have in a season. This will be accomplished by splitting the data into a training set to develop the model and a validation set to determine how well the models perform. Several approaches will be applied including manually selecting predictor variables, Stepwise selection and Decision Tree selection using Angoss Knowledge Studio. The best model will be selected and tested against a holdout sample where the number of target wins is unknown by the author. The performance of the model against the holdout sample will determine if the model is adequate for real world prediction.

Data Exploration

The first step necessary in building our model is to explore the data available starting with what we are trying to predict: the number of wins in a season for baseball teams. By performing univariate analysis on TARGET_WINS we see that it has a fairly normal distribution which means we should be able to fit a model that is fairly accurate in predicting team performance. The probability and distribution plot does reveal that there are some outliers which is why the data points tail off on the TARGET_WINS versus normal quantiles plot.



Also knowing the mean and median number of wins based on historical data is important. This gives us a metric to determine if our model is performing well. At a minimum, a good model should perform better than assuming 82 wins for each team which is the median number of wins from 1871 to 2006.

Basic Statistical Measures			
Location		Variability	
Mean	80.79086	Std Deviation	15.75215
Median	82.00000	Variance	248.13031
Mode	83.00000	Range	146.00000
		Interquartile Range	21.00000

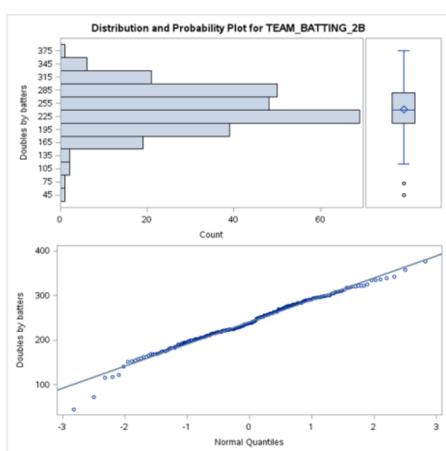
Correlation of Team Statistics to Target Wins

Another important aspect of predicting the number of wins a team will have in as season is to understand how the data available relates to this outcome. To accomplish this I created a table using the Pearson correlation coefficient. The higher the value of the coefficient, the more the data should impact TARGET_WINS and the sign indicates if it will be an increase or decrease. So for example TEAM_BATTING_H has a coefficient of 0.3887 which means it has a strong positive correlation with TARGET_WINS. The PCC table was integral in selecting which data to include in my first model.

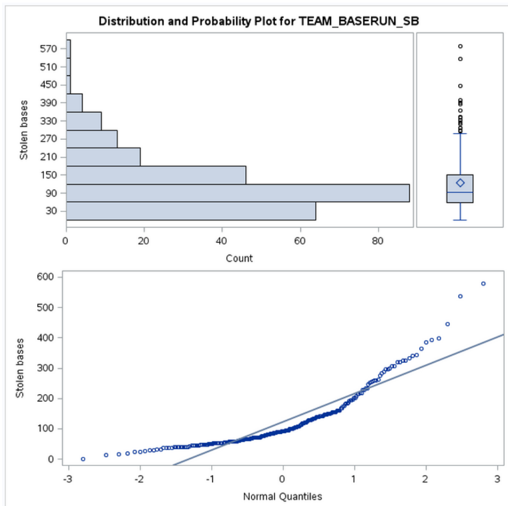
Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations	TARGET_WINS
TEAM_BATTING_H Base Hits by batters	0.38877 <.0001 2276
TEAM_BATTING_2B Doubles by batters	0.28910 <.0001 2276
TEAM_BATTING_3B Triples by batters	0.14261 <.0001 2276
TEAM_BATTING_HR Homeruns by batters	0.17615 <.0001 2276
TEAM_BATTING_BB Walks by batters	0.23256 <.0001 2276
TEAM_BATTING_SO Strikeouts by batters	-0.03175 0.1389 2174
TEAM_BASERUN_SB Stolen bases	0.13514 <.0001 2145
TEAM_BASERUN_CS Caught stealing	0.02240 0.3853 1504
TEAM_BATTING_HBP Batters hit by pitch	0.07350 0.3122 191
TEAM_PITCHING_H Hits allowed	-0.10994 <.0001 2276
TEAM_PITCHING_HR Homeruns allowed	0.18901 <.0001 2276
TEAM_PITCHING_BB Walks allowed	0.12417 <.0001 2276
TEAM_PITCHING_SO Strikeouts by pitchers	-0.07844 0.0003 2174
TEAM_FIELDING_E Errors	-0.17648 <.0001 2276
TEAM_FIELDING_DP Double Plays	-0.03485 0.1201 1990

Exploration of Potential Predictors

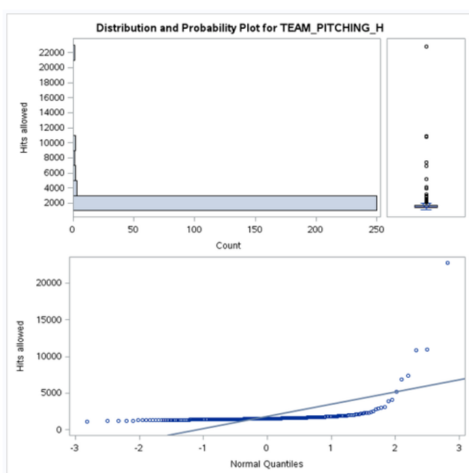
Next I explored the available data to determine if there would be issues with missing values an outliers. An example of data that can be used with little concern would be TEAM_BATTING_2B which is the number of doubles a team hits in a season although the Distribution and Probability plot below show some outliers, the quantile plot include is fairly well fit indicating solid, normally distributed data.



In contrast the Distribution and Probability plot for TEAM_BASERUN_SB (Stolen Bases) is heavily skewed with several teams having an extremely high number of stolen bases compared to the rest of the teams. The strong curve in the quantile plot further supports the concern that there are outliers here. This warrants further exploration and it may be necessary to perform a transformation so that the outliers do not negatively influence any regression models fitted from the data.



An extreme example of outliers can be found in TEAM_PITCHING_H where some teams have numbers in the 10,000-20,000 range. This is likely caused by normalizing older data in a time when teams played a fraction of the games in a season as they do today. This must definitely be addressed if we are to use this variable as a predictor.



This same analysis was performed on all potential predictor variables to help identify all outliers and to determine whether or not the predictors are normally distributed. The remaining results have been omitted for the sake of brevity.

Lastly, I took a snapshot of the data to better understand average, extreme and missing values in the data set. This was used during data preparation to determine which variables need to be transformed before fitting our models.

Exploration of Historical Team Data

The MEANS Procedure

Variable	Label	Mean	Median	Minimum	Maximum	1st Pctl	10th Pctl	90th Pctl	99th Pctl	N	N Miss
TARGET_WINS		80.79	82.00	0.00	146.00	38.00	61.00	100.00	114.00	2276	0
TEAM_BATTING_H	Base Hits by batters	1469.27	1454.00	891.00	2554.00	1188.00	1315.00	1636.00	1950.00	2276	0
TEAM_BATTING_2B	Doubles by batters	241.25	238.00	69.00	458.00	141.00	182.00	303.00	352.00	2276	0
TEAM_BATTING_3B	Triples by batters	55.25	47.00	0.00	223.00	17.00	27.00	96.00	134.00	2276	0
TEAM_BATTING_HR	Homeruns by batters	99.61	102.00	0.00	264.00	4.00	20.00	180.00	235.00	2276	0
TEAM_BATTING_BB	Walks by batters	501.56	512.00	0.00	878.00	79.00	363.00	635.00	755.00	2276	0
TEAM_BATTING_SO	Strikeouts by batters	735.61	750.00	0.00	1399.00	67.00	421.00	1049.00	1193.00	2174	102
TEAM_BASERUN_SB	Stolen bases	124.76	101.00	0.00	697.00	23.00	44.00	231.00	439.00	2145	131
TEAM_BASERUN_CS	Caught stealing	52.80	49.00	0.00	201.00	16.00	30.00	77.00	143.00	1504	772
TEAM_BATTING_HBP	Batters hit by pitch	59.36	58.00	29.00	95.00	29.00	44.00	76.00	90.00	191	2085
TEAM_PITCHING_H	Hits allowed	1779.21	1518.00	1137.00	30132.00	1244.00	1356.00	2059.00	7093.00	2276	0
TEAM_PITCHING_HR	Homeruns allowed	105.70	107.00	0.00	343.00	8.00	25.00	187.00	244.00	2276	0
TEAM_PITCHING_BB	Walks allowed	553.01	536.50	0.00	3645.00	237.00	417.00	694.00	924.00	2276	0
TEAM_PITCHING_SO	Strikeouts by pitchers	817.73	813.50	0.00	19278.00	205.00	490.00	1095.00	1474.00	2174	102
TEAM_FIELDING_E	Errors	246.48	159.00	65.00	1898.00	86.00	109.00	542.00	1237.00	2276	0
TEAM_FIELDING_DP	Double Plays	146.39	149.00	52.00	228.00	79.00	109.00	178.00	204.00	1990	286

DATA PREPARATION

The first concern that arises from the Exploration of Historical Team Data table is that the TEAM_BATTING_HBP variable is missing a significant number of observations. I determined that there are too many values missing to accurately use imputation as a method to correct this so this variable will be removed from consideration.

Next, it was necessary to fix the remaining variables with missing values. These include TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_PITCHING_SO and TEAM_FIELDING_DP. The method selected to fix these variables was to replace missing values with the mean from the Exploration of Historical Team Data table. Because the fact that a value is missing can be predictive, flag variables were created to track when values were imputed.

As discussed in the Data Exploration section, some variables such as TEAM_PITCHING_H have extreme values that will influence our models. In cases where there were extreme values, they were replaced with the 90th percentile value to ensure normalized values or bad data from the data set were accommodated for. In the case of TEAM_PITCHING_H, it is not possible to give up 30,000 hits in a season so this outlier must be dealt with. Other variable were evaluated on the same grounds. If the value is not realistic for a team to achieve in a season, it was replaced with the 90th percentile value.

Lastly there were two variables that I calculated from the existing data and added to the data set. The first calculated variable was base hits which I named TEAM_BATTING_1B. This was calculated by taking TEAM_BATTING_H which represents the total hits for the season and subtracting doubles, triples and homeruns. As a result we now have a complete breakdown of all successful at bats and we now have the ability to determine any or all are predictive of team wins. Since TEAM_BATTING_H is a linear combination of singles, doubles, triples and home runs, I did not consider it in my models as that would be redundant.

The next calculated variable was Stolen Base percentage which was calculated from TEAM_BASERUN_SB and TEAM_BASERUN_CS. This gives us a metric as to how successful a team is at stealing bases and in theory, more stolen bases should result in more wins.

Lastly, the transformed data was split into a training set to fit potential models and a validation set to determine how well the models predict how many wins the teams will have in a season. Below is a snapshot of the transformed training data. We can see there are no longer any missing values and outliers have been replaced with more realistic values. This is also true of the validation data set.

Exploration of MBTrain Data

The MEANS Procedure

Variable	Label	Mean	Median	1st Pctl	10th Pctl	90th Pctl	99th Pctl	Minimum	Maximum	N	N Miss
INDEX		1287.89	1315.00	28.00	257.00	2313.00	2512.00	1.00	2535.00	1599	0
TARGET_WINS		80.51	82.00	38.00	60.00	99.00	116.00	0.00	146.00	1599	0
TEAM_BATTING_H	Base Hits by batters	1469.52	1454.00	1187.00	1311.00	1637.00	1978.00	891.00	2554.00	1599	0
TEAM_BATTING_2B	Doubles by batters	241.24	239.00	141.00	180.00	303.00	352.00	113.00	393.00	1599	0
TEAM_BATTING_3B	Triples by batters	55.46	47.00	27.00	27.00	95.00	133.00	27.00	223.00	1599	0
TEAM_BATTING_HR	Homeruns by batters	100.97	104.00	20.00	20.00	179.00	235.00	20.00	264.00	1599	0
TEAM_BATTING_BB	Walks by batters	510.90	508.00	363.00	363.00	631.00	734.00	363.00	878.00	1599	0
TEAM_BATTING_SO	Strikeouts by batters	750.32	739.00	421.00	421.00	1049.00	1191.00	421.00	1399.00	1599	0
TEAM_BASERUN_SB	Stolen bases	124.08	105.00	44.00	45.00	221.00	429.00	44.00	697.00	1599	0
TEAM_BASERUN_CS	Caught stealing	53.12	53.00	30.00	33.00	72.00	123.00	30.00	201.00	1599	0
TEAM_PITCHING_H	Hits allowed	1582.64	1518.00	1244.00	1353.00	2059.00	2059.00	1137.00	2059.00	1599	0
TEAM_PITCHING_HR	Homeruns allowed	107.59	109.00	25.00	25.00	188.00	249.00	25.00	343.00	1599	0
TEAM_PITCHING_BB	Walks allowed	543.19	533.00	417.00	420.00	694.00	694.00	417.00	694.00	1599	0
TEAM_PITCHING_SO	Strikeouts by pitchers	800.81	818.00	490.00	494.00	1095.00	1095.00	490.00	1095.00	1599	0
TEAM_FIELDING_E	Errors	218.31	157.00	84.00	109.00	542.00	542.00	65.00	542.00	1599	0
TEAM_FIELDING_DP	Double Plays	146.16	146.00	79.00	111.00	177.00	201.00	52.00	225.00	1599	0
SB_PCT		0.65	0.63	0.63	0.63	0.72	0.79	0.63	0.84	1599	0
FLAG_SB_FIXED		0.33	0.00	0.00	0.00	1.00	1.00	0.00	1.00	1599	0
FLAG_BATSO_FIXED		0.05	0.00	0.00	0.00	0.00	1.00	0.00	1.00	1599	0
FLAG_BRSB_FIXED		0.06	0.00	0.00	0.00	0.00	1.00	0.00	1.00	1599	0
FLAG_BRCS_FIXED		0.33	0.00	0.00	0.00	1.00	1.00	0.00	1.00	1599	0
FLAG_FDP_FIXED		0.12	0.00	0.00	0.00	1.00	1.00	0.00	1.00	1599	0
FLAG_PSO_FIXED		0.05	0.00	0.00	0.00	0.00	1.00	0.00	1.00	1599	0
TEAM_BATTING_1B		1071.85	1048.00	874.00	939.00	1224.00	1633.00	682.00	2107.00	1599	0

BUILD MODELS

Model 1

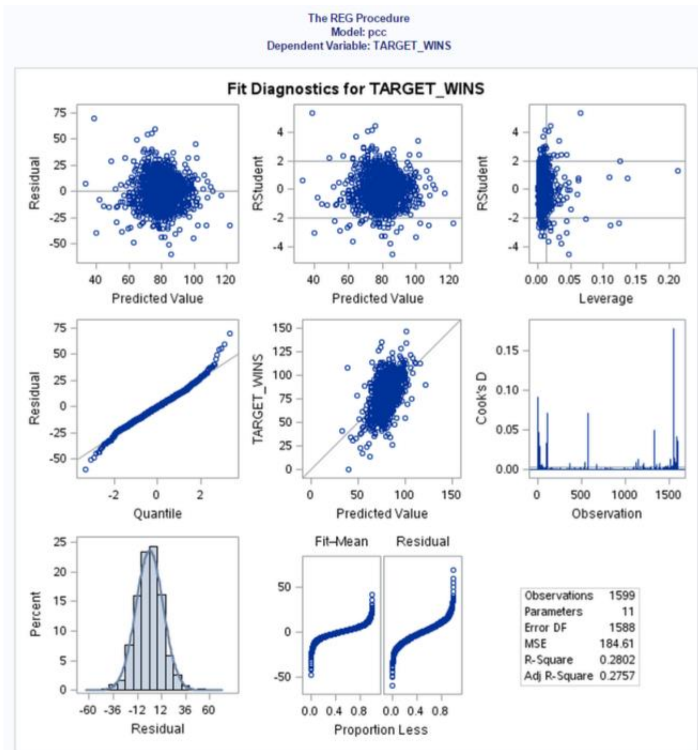
The first model that was evaluated was based on manual selection with the aid of the Pearson Correlation Coefficients of the variables from the original data set. A table of these values can be found in the Data Exploration section. Higher numbers indicate a stronger relationship to the response variable, TARGET_WINS so in theory, selecting variables with higher PCC values should produce a better model. My selection criteria were variables with PCC values greater than .10. This resulted in the following model after regression:

TARGET_WINS = 4.53146
+ 0.0373*TEAM_BATTING_1B
+ 0.01415*TEAM_BATTING_2B
+ 0.13823*TEAM_BATTING_3B
+ 0.05439*TEAM_BATTING_HR
+ 0.058*TEAM_BATTING_BB
+ 0.04574*TEAM_BASERUN_SB
+ 0.00962*TEAM_PITCHING_H
+ 0.00659*TEAM_PITCHING_HR
+ -0.04083*TEAM_PITCHING_BB
+ -0.04412*TEAM_FIELDING_E;

Parameter Estimates							
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	4.53146	4.30107	1.05	0.2922	0
TEAM_BATTING_1B		1	0.03730	0.00393	9.50	<.0001	2.36879
TEAM_BATTING_2B	Doubles by batters	1	0.01415	0.00992	1.43	0.1541	1.87506
TEAM_BATTING_3B	Triples by batters	1	0.13823	0.01986	6.96	<.0001	2.46845
TEAM_BATTING_HR	Homeruns by batters	1	0.05439	0.03075	1.77	0.0771	29.31645
TEAM_BATTING_BB	Walks by batters	1	0.05800	0.01108	5.24	<.0001	9.02382
TEAM_BASERUN_SB	Stolen bases	1	0.04574	0.00538	8.50	<.0001	1.66018
TEAM_PITCHING_H	Hits allowed	1	0.00962	0.00322	2.98	0.0029	4.62004
TEAM_PITCHING_HR	Homeruns allowed	1	0.00659	0.02732	0.24	0.8095	23.93071
TEAM_PITCHING_BB	Walks allowed	1	-0.04083	0.01110	-3.68	0.0002	8.09189
TEAM_FIELDING_E	Errors	1	-0.04412	0.00510	-8.64	<.0001	4.32677

When looking at the Parameter Estimates, the model makes sense for the most part but there are concerns with multicollinearity and two variables have parameters with signs that are opposite of what we would expect. The positive sign for TEAM_PITCHING_H and TEAM_PITCHING_HR indicates the model rewards teams for allowing hits and homeruns. This is likely a result of the variables I selected having a correlation to each other as indicated by the Variance Inflation (VIF) values. Values greater than 10 are a red flag that this is occurring.

The primary focus for the Fit Diagnostics graphics below is for the TARGET_WINS vs Predicted Value plot in the center. The plots in a good model should follow the diagonal line and we can see this pattern generally occurring so the model has decent predictive power.



Model 2

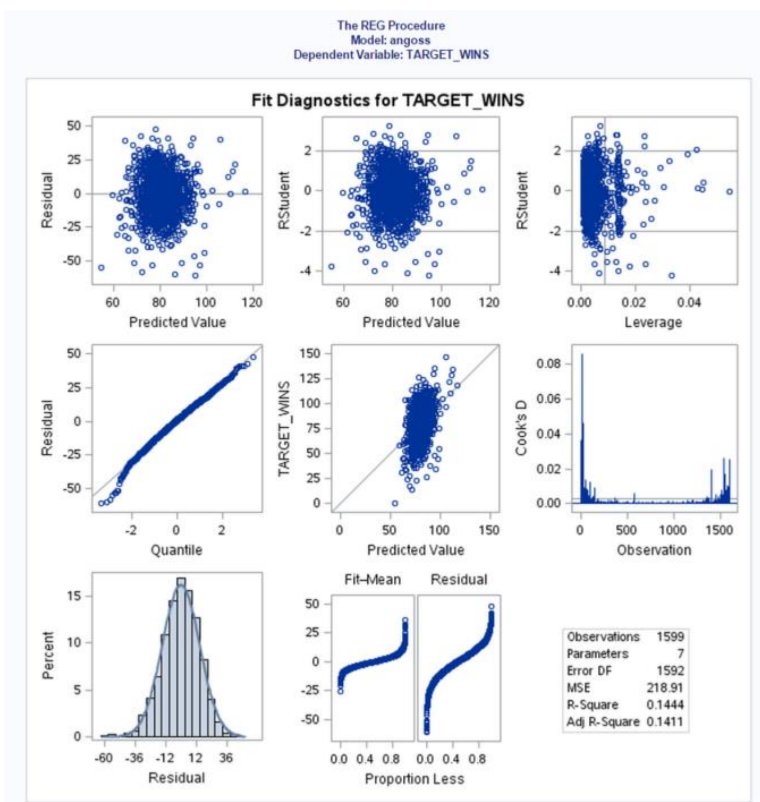
The next model evaluated was I created by using the variables from a decision tree in Angoss KnowledgeStudio (screenshots are in Bonus Bingo section). The MBTraining data set was fed into Angoss and the decision tree identified certain variables as predictive in regards to TARGET_WINS. After running a regression analysis on the variables selected by Angoss, the following equation resulted:

TARGET_WINs = 27.49448
+ 0.03067*TEAM_BATTING_1B
+ 0.08994*TEAM_BATTING_2B
+ -0.01648*TEAM_BASERUN_CS

+ -0.98614*FLAG_SB_FIXED
+ 6.12658*FLAG_BATSO_FIXED
+ -5.2505*FLAG_FDP_FIXED

Parameter Estimates							
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	27.49448	3.63149	7.57	<.0001	0
TEAM_BATTING_1B		1	0.03067	0.00335	9.17	<.0001	1.45130
TEAM_BATTING_2B	Doubles by batters	1	0.08994	0.00884	10.18	<.0001	1.25424
TEAM_BASERUN_CS	Caught stealing	1	-0.01648	0.02077	-0.79	0.4277	1.04448
FLAG_SB_FIXED		1	-0.98614	1.03558	-0.95	0.3411	1.72749
FLAG_BATSO_FIXED		1	6.12658	1.97690	3.10	0.0020	1.25995
FLAG_FDP_FIXED		1	-5.25050	1.41547	-3.71	0.0002	1.56706

At first glance this appears to be a good model. The variable parameter signs all makes sense and the low VIF values indicate there a no issues with multicollinearity. However when we inspect the Fit Diagnostics from PROC REG, we see that the TARGET_WINS versus Predicted Value plot in the center indicates the model may not be so good. The plots are nearly vertical which tells us the model is not predicting wins very accurately.



Model 3

For the final model, I let SAS automatically select which variables should be included in the predictive model. Any variable that was not discarded during Data Preparation was considered and Stepwise selection was applied. The Stepwise selection process allows the most possible combinations of variables which is why this method was chosen. This resulted in the following model:

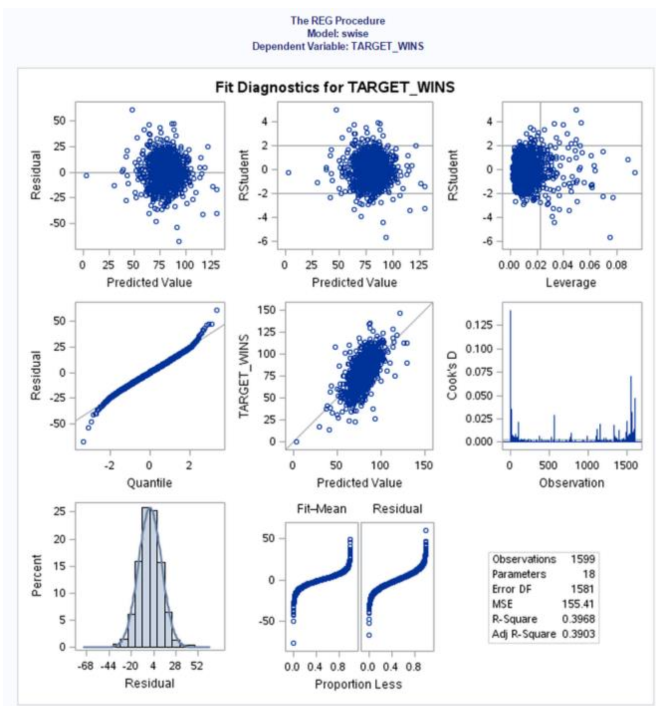
TARGET_WINS = 56.41909
+ 0.05178*TEAM_BATTING_1B
+ 0.0333*TEAM_BATTING_2B
+ 0.16274*TEAM_BATTING_3B
+ 0.10789*TEAM_BATTING_HR
+ 0.03136*TEAM_BATTING_BB
+ -0.02736*TEAM_BATTING_SO
+ 0.07572*TEAM_BASERUN_SB
+ -0.04694*TEAM_BASERUN_CS
+ -0.11258*TEAM_FIELDING_E
+ -0.11187*TEAM_FIELDING_DP
+ -0.01241*TEAM_PITCHING_H
+ 0.01353*TEAM_PITCHING_SO
+ -24.63666*SB_PCT
+ 7.63914*FLAG_BATSO_FIXED
+ 31.29647*FLAG_BRSB_FIXED
+ 2.4961*FLAG_BRCS_FIXED
+ 8.98023*FLAG_FDP_FIXED

Upon initial inspection this model has many concerns. The VIF values suggest several multicollinearity issues and many of the parameter signs do not make sense. It also appears that all variables were included with exception of two flag variables. It seems highly unlikely that all variables would be predictive of TARGET_WINS.

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	56.41909	9.30280	6.06	<.0001	0
TEAM_BATTING_1B		1	0.05178	0.00410	12.64	<.0001	3.06559
TEAM_BATTING_2B	Doubles by batters	1	0.03330	0.00931	3.58	0.0004	1.96124
TEAM_BATTING_3B	Triples by batters	1	0.16274	0.01921	8.47	<.0001	2.74276
TEAM_BATTING_HR	Homeruns by batters	1	0.10789	0.01107	9.75	<.0001	4.51243
TEAM_BATTING_BB	Walks by batters	1	0.03136	0.00428	7.33	<.0001	1.60094
TEAM_BATTING_SO	Strikeouts by batters	1	-0.02736	0.00606	-4.51	<.0001	18.49330
TEAM_BASERUN_SB	Stolen bases	1	0.07572	0.00678	11.16	<.0001	3.13152
TEAM_BASERUN_CS	Caught stealing	1	-0.04694	0.02129	-2.20	0.0276	1.54516
TEAM_FIELDING_E	Errors	1	-0.11258	0.00744	-15.13	<.0001	10.92229
TEAM_FIELDING_DP	Double Plays	1	-0.11187	0.01691	-6.62	<.0001	1.81698
TEAM_PITCHING_H	Hits allowed	1	-0.01241	0.00301	-4.12	<.0001	4.78917
TEAM_PITCHING_SO	Strikeouts by pitchers	1	0.01353	0.00564	2.40	0.0165	12.86883
SB_PCT		1	-24.63666	10.10398	-2.44	0.0149	1.72350
FLAG_BATSO_FIXED		1	7.63914	1.86003	4.11	<.0001	1.57110
FLAG_BRSB_FIXED		1	31.29647	2.27467	13.76	<.0001	2.91615
FLAG_BRCS_FIXED		1	2.49610	1.20384	2.07	0.0383	3.28505
FLAG_FDP_FIXED		1	8.98023	1.99237	4.51	<.0001	4.37325

In contrast to my initial inspection, the TARGET_WINS versus Predicted Value plot tells another story. The plots adhere very well to the diagonal line indicating that despite concerns with multicollinearity and counterintuitive parameter signs, the model is very good a predicting TARGET_WINS.



SELECT MODELS

Model selection was determined using a two-pronged approach. The first criteria applied were RMSE, AIC and BIC values that were captured during the regression analysis step within SAS. Models with lower values are considered to have more accurate predictive abilities and as demonstrated in the table below, the model fitted using the Stepwise (swise) selection scored best in all three areas.

Comparison of Models Based on Key Statistics

Obs	_MODEL_	_RMSE_	_AIC_	_BIC_
1	swise	12.4665	8086.59	8088.97
2	pcc	13.5873	8354.98	8357.13
3	angoss	14.7957	8623.47	8625.53

The next test used to select the best model was a squared error calculation. The generic formula being:

MODEL_ERROR = (TARGET_WINS - PREDICTED_WINS)**2

This formula was applied to all observations in the MBValidation data set for each model and ranked by Mean and Median values. Again, lower values are better. Also, to serve as a reference point this same calculation was performed using the TARGET_WINS median value of 82 and was included in the results under the variable ERROR_AVE_WINS. This gives us some indication as to whether or not the models are performing better than average. As indicated by the table below, the Stepwise model wins here too.

Squared Error Comparison of Models Using Validation Data

The MEANS Procedure

Variable	Mean	Median
ERROR_My_Selection	162.3590426	69.4063941
ERROR_Angoss	208.0185561	91.4890337
ERROR_Stepwise	129.1031916	51.5132427
ERROR_AVE_WINS	231.8788774	100.0000000

Based on RMSE, AIC, BIC from the training data and the Squared Error formula from the validation data, the Stepwise model is the best predictor. Adjusted R-squared could have also been included as a deciding factor but it would not have changed the outcome, the Stepwise model again exceeded the others in this category

Winner: Stepwise Automated Variable Selection Model.

Conclusion:

Three models were evaluated and the Stepwise model performed best based on the selected test metrics. This selection is not without concerns. There are issues with multicollinearity in the Stepwise model and some parameter estimate signs are not in agreement with logic. As an example, stolen base percentage has a negative impact on wins meaning the more successful a team is at stealing bases, the more losses they should have in a season. This is counterintuitive and if it were not for the model's strong performance in the Squared Error comparison, a different model would have been selected. Further analysis is required and it is likely that resolving the multicollinearity issues would also correct the parameter estimate signs. Lastly, because there is no such thing as a perfect model, P_TARGET_WINS was limited to a range of 50-105 to ensure that results represent the number of games a modern baseball team can realistically be expected to win.

Code:

Code for EDA and Model Selection:

```
%let PATH =/folders/myfolders/sasuser.v94/411_data/Moneyball;
%let NAME = MB;
%let LIB = &NAME..;

libname &NAME. "&PATH.";

%let INFILE = &LIB.MONEYBALL;

*Explore response variable;

proc univariate data=&INFILE. plot;
	var TARGET_WINS;
run;


* Check correlations among variables with TARGET_WINS;

ods graphics on;
proc corr data=&INFILE. ;
VAR TARGET_WINS;
WITH TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
```

```
TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP;
TITLE "Correlation TARGET_WINS Vs. Predictor Variables";
run;
ods graphics off;

* Univariate analysis predictors with Pearson CC > .10;

proc univariate data=&INFILE. plot;
	var TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR TEAM_BATTING_BB
		TEAM_BASERUN_SB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
		TEAM_FIELDING_E;
TITLE "Univariate Analysis Variables with Peaarson Correlation Coefficient > .10";
run;


*Explore missing values in data set;

data data_exploration;
	set &INFILE.;
	drop index;
run;

proc means data=data_exploration mean median min max p1 p10 p90 p99 max ndec=2 n nmiss;
TITLE "Exploration of Historical Team Data";
run;

* Impute missing data using mean and fix outliers;

data TEMP1;
set &INFILE.;

drop TEAM_BATTING_HBP;

SB_PCT = TEAM_BASERUN_SB /(TEAM_BASERUN_SB + TEAM_BASERUN_CS );

FLAG_SB_FIXED = missing(SB_PCT);
FLAG_BATSO_FIXED = missing(TEAM_BATTING_SO);
FLAG_BRSB_FIXED = missing(TEAM_BASERUN_SB);
FLAG_BRCS_FIXED = missing(TEAM_BASERUN_CS);
FLAG_FDP_FIXED = missing(TEAM_FIELDING_DP);
FLAG_PSO_FIXED = missing(TEAM_PITCHING_SO);

if missing(TEAM_BATTING_SO) then TEAM_BATTING_SO = 736;
```

```
if missing(TEAM_BASERUN_SB) then TEAM_BASERUN_SB = 125;
if missing(TEAM_BASERUN_CS) then TEAM_BASERUN_CS = 53;
if missing(TEAM_FIELDING_DP) then TEAM_FIELDING_DP = 146;
if missing(TEAM_PITCHING_SO) then TEAM_PITCHING_SO = 818;
if missing(SB_PCT) then SB_PCT = 0.63;

if TEAM_FIELDING_E > 542 then TEAM_FIELDING_E = 542;
if TEAM_PITCHING_BB > 694 then TEAM_PITCHING_BB = 694;
if TEAM_PITCHING_H > 2059 then TEAM_PITCHING_H = 2059;
if TEAM_PITCHING_SO > 1095 then TEAM_PITCHING_SO = 1095;


if SB_PCT < 0.63 then SB_PCT = 0.63;
if TEAM_PITCHING_BB < 417 then TEAM_PITCHING_BB = 417;
if TEAM_PITCHING_HR < 25 then TEAM_PITCHING_HR = 25;
if TEAM_PITCHING_SO < 490 then TEAM_PITCHING_SO = 490;
if TEAM_BATTING_SO < 421 then TEAM_BATTING_SO = 421;
if TEAM_BASERUN_SB < 44 then TEAM_BASERUN_SB = 44;
if TEAM_BASERUN_CS < 30 then TEAM_BASERUN_CS = 30;
if TEAM_BATTING_3B < 27 then TEAM_BATTING_3B = 27;
if TEAM_BATTING_HR < 20 then TEAM_BATTING_HR = 20;
if TEAM_BATTING_BB < 363 then TEAM_BATTING_BB = 363;

TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B -
TEAM_BATTING_HR;

run;

* Split moneyball data set into training and validation sets using a 70/30 split;

data MBTrain MBValidate;
set TEMP1;
if ranuni(1) < 0.70 then
	output MBTrain;
else
	output MBValidate;
run;

* Export MBTrain to create Decision Tree in Angoss;

proc export data=MBTrain file='/folders/myfolders/sasuser.v94/411_data/Moneyball/MBTrain.csv'
replace;
run;
```

```
* Spot check training and validation data;

proc print data = MBTrain (obs = 10);
run;

proc print data = MBValidate (obs = 10);
run;

* Check training and validation data for remaing outliers and missing values;

proc means data=MBTrain mean median p1 p10 p90 p99 min max ndec=2 n nmiss;
TITLE "Exploration of MBTrain Data";
run;


proc means data=MBValidate mean median p1 p10 p90 p99 min max ndec=2 n nmiss;
TITLE "Exploration of MBValidata Data";
run;


/* Below are the 3 models selected for comparison. The first is a model I selected using variables with a
Pearson */
/* correlation coefficient greater that .10. The second model was selected by feeding the MBTrain
dataset into Angoss and */
/* selecting variables using a decision tree. The final model was selected by starting with all variables
except */
/* TEAM_BATTING_H (Since this is a linear combination of 1B,2B,3B and HR) and applying stepwise
variable selection with */
/* PROC REG. */

* My selection based on PCC .1 or greater;

proc reg data = MBTrain OUTEST = est1;
	pcc: model TARGET_WINS =
		TEAM_BATTING_1B
		TEAM_BATTING_2B
		TEAM_BATTING_3B
		TEAM_BATTING_HR
		TEAM_BATTING_BB
		TEAM_BASERUN_SB
		TEAM_PITCHING_H
		TEAM_PITCHING_HR
		TEAM_PITCHING_BB
		TEAM_FIELDING_E/ vif aic bic;
```

```
run;
quit;


* Model selected by Decision Tree in Angoss;

proc reg data = MBTrain OUTEST = est2;
	angoss: model TARGET_WINs =
				TEAM_BATTING_1B
				TEAM_BATTING_2B
				TEAM_BASERUN_CS
				FLAG_SB_FIXED
				FLAG_BATSO_FIXED
				FLAG_FDP_FIXED/ vif aic bic;
run;
quit;

* All variables using stepwise selection;

proc reg data=MBTrain outest = est3;
	swise: model TARGET_WINS =
				TEAM_BATTING_1B
				TEAM_BATTING_2B
				TEAM_BATTING_3B
				TEAM_BATTING_HR
				TEAM_BATTING_BB
				TEAM_BATTING_SO
				TEAM_BASERUN_SB
				TEAM_BASERUN_CS
				TEAM_FIELDING_E
				TEAM_FIELDING_DP
				TEAM_PITCHING_BB
				TEAM_PITCHING_H
				TEAM_PITCHING_HR
				TEAM_PITCHING_SO
				SB_PCT FLAG_SB_FIXED
				FLAG_BATSO_FIXED
				FLAG_BRSB_FIXED
				FLAG_BRCS_FIXED
				FLAG_FDP_FIXED
				FLAG_PSO_FIXED/ selection = stepwise vif aic bic;
run;
quit;
```

```
* Compare data sets add out files for other models;

data estout;
	set est3 est2 est1;
	keep _MODEL_ _RMSE_ _AIC_ _BIC_;
	run;
	proc sort data=estout; by _AIC_;
proc print data=estout;
TITLE "Comparison of Models Based on Key Statistics";
run;


*Score Code Below;

%let results = MBValidate;

data score_file;
set &results.;

* UNCOMMENT THE TRANSFORMATIONS WHEN CREATING FINAL SCORING DATA STEP, NO NEED TO DO
THIS HERE, MBValidate ALREADY TRANSFORMED;

/* drop TEAM_BATTING_HBP; */
/*  */
/* SB_PCT = TEAM_BASERUN_SB /(TEAM_BASERUN_SB + TEAM_BASERUN_CS ); */
/*  */
/* FLAG_SB_FIXED = missing(SB_PCT); */
/* FLAG_BATSO_FIXED = missing(TEAM_BATTING_SO); */
/* FLAG_BRSB_FIXED = missing(TEAM_BASERUN_SB); */
/* FLAG_BRCS_FIXED = missing(TEAM_BASERUN_CS); */
/* FLAG_FDP_FIXED = missing(TEAM_FIELDING_DP); */
/* FLAG_PSO_FIXED = missing(TEAM_PITCHING_SO); */
/*  */
/* if missing(TEAM_BATTING_SO) then TEAM_BATTING_SO = 736;  */
/* if missing(TEAM_BASERUN_SB) then TEAM_BASERUN_SB = 125; */
/* if missing(TEAM_BASERUN_CS) then TEAM_BASERUN_CS = 53;  */
/* if missing(TEAM_FIELDING_DP) then TEAM_FIELDING_DP = 146; */
/* if missing(TEAM_PITCHING_SO) then TEAM_PITCHING_SO = 818; */
/* if missing(SB_PCT) then SB_PCT = 0.63; */
/*  */
/* if TEAM_FIELDING_E > 542 then TEAM_FIELDING_E = 542; */
/* if TEAM_PITCHING_BB > 694 then TEAM_PITCHING_BB = 694; */
```

```
/* if TEAM_PITCHING_H > 2059 then TEAM_PITCHING_H = 2059; */
/* if TEAM_PITCHING_SO > 1095 then TEAM_PITCHING_SO = 1095; */
/*  */
/*  */
/* if SB_PCT < 0.63 then SB_PCT = 0.63; */
/* if TEAM_PITCHING_BB < 417 then TEAM_PITCHING_BB = 417; */
/* if TEAM_PITCHING_HR < 25 then TEAM_PITCHING_HR = 25; */
/* if TEAM_PITCHING_SO < 490 then TEAM_PITCHING_SO = 490; */
/* if TEAM_BATTING_SO < 421 then TEAM_BATTING_SO = 421; */
/* if TEAM_BASERUN_SB < 44 then TEAM_BASERUN_SB = 44; */
/* if TEAM_BASERUN_CS < 30 then TEAM_BASERUN_CS = 30; */
/* if TEAM_BATTING_3B < 27 then TEAM_BATTING_3B = 27; */
/* if TEAM_BATTING_HR < 20 then TEAM_BATTING_HR = 20; */
/* if TEAM_BATTING_BB < 363 then TEAM_BATTING_BB = 363; */
/*  */
/* TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B -
TEAM_BATTING_HR; */

/*run;*/


*My equation based on PCC > .10 ;
PRED_1 = 4.53146
                  + 0.0373*TEAM_BATTING_1B
                  + 0.01415*TEAM_BATTING_2B
                  + 0.13823*TEAM_BATTING_3B
                  + 0.05439*TEAM_BATTING_HR
                  + 0.058*TEAM_BATTING_BB
                  + 0.04574*TEAM_BASERUN_SB
                  + 0.00962*TEAM_PITCHING_H
                  + 0.00659*TEAM_PITCHING_HR
                  + -0.04083*TEAM_PITCHING_BB
                  + -0.04412*TEAM_FIELDING_E;


* Variables selected by Angoss Decision Tree;


PRED_2 = 27.49448
                           + 0.03067*TEAM_BATTING_1B
                           + 0.08994*TEAM_BATTING_2B
                           + -0.01648*TEAM_BASERUN_CS
                           + -0.98614*FLAG_SB_FIXED
                           + 6.12658*FLAG_BATSO_FIXED
```

```
                                + -5.2505*FLAG_FDP_FIXED;


* Equation using all variables with stepwise selection;

PRED_3 = 56.41909
                                + 0.05178*TEAM_BATTING_1B
                                + 0.0333*TEAM_BATTING_2B
                                + 0.16274*TEAM_BATTING_3B
                                + 0.10789*TEAM_BATTING_HR
                                + 0.03136*TEAM_BATTING_BB
                                + -0.02736*TEAM_BATTING_SO
                                + 0.07572*TEAM_BASERUN_SB
                                + -0.04694*TEAM_BASERUN_CS
                                + -0.11258*TEAM_FIELDING_E
                                + -0.11187*TEAM_FIELDING_DP
                                + -0.01241*TEAM_PITCHING_H
                                + 0.01353*TEAM_PITCHING_SO
                                + -24.63666*SB_PCT
                                + 7.63914*FLAG_BATSO_FIXED
                                + 31.29647*FLAG_BRSB_FIXED
                                + 2.4961*FLAG_BRCS_FIXED
                                + 8.98023*FLAG_FDP_FIXED;


AVE_WINS_DIFF = (TARGET_WINS - 82);

if PRED_1 < 50  then PRED_1 = 50;
if PRED_1 > 105 then PRED_1 = 105;

if PRED_2 < 50  then PRED_2 = 50;
if PRED_2 > 105 then PRED_2 = 105;

if PRED_3 < 50  then PRED_3 = 50;
if PRED_3 > 105 then PRED_3 = 105;


PRED_1_DIFF = (TARGET_WINS - PRED_1);
PRED_2_DIFF = (TARGET_WINS - PRED_2);
PRED_3_DIFF = (TARGET_WINS - PRED_3);

ERROR_My_Selection = (TARGET_WINS - PRED_1)**2;
ERROR_Angoss = (TARGET_WINS - PRED_2)**2;
ERROR_Stepwise = (TARGET_WINS - PRED_3)**2;
```

```
ERROR_AVE_WINS = AVE_WINS_DIFF **2;

run;

proc print data = score_file;
var TARGET_WINS PRED_1 PRED_2 PRED_3 PRED_1_DIFF PRED_2_DIFF
 PRED_3_DIFF AVE_WINS_DIFF ERROR_My_Selection ERROR_Angoss ERROR_Stepwise
ERROR_AVE_WINS;
run;

proc means data=score_file mean median;
var ERROR_My_Selection ERROR_Angoss ERROR_Stepwise ERROR_AVE_WINS;
TITLE "Squared Error Comparison of Models Using Validation Data";
run;
```

Code for Scored File Data Step:

```
%let PATH =/folders/myfolders/sasuser.v94/411_data/Moneyball;
%let NAME = MB;
%let LIB = &NAME..;

libname &NAME. "&PATH.";

%let INFILE = &LIB.MONEYBALL_TEST;

data moneyball_test_scores;
set &INFILE.;

* Perform the same transformations that were applied to training data;

drop TEAM_BATTING_HBP;

SB_PCT = TEAM_BASERUN_SB /(TEAM_BASERUN_SB + TEAM_BASERUN_CS );

FLAG_SB_FIXED = missing(SB_PCT);
FLAG_BATSO_FIXED = missing(TEAM_BATTING_SO);
FLAG_BRSB_FIXED = missing(TEAM_BASERUN_SB);
FLAG_BRCS_FIXED = missing(TEAM_BASERUN_CS);
FLAG_FDP_FIXED = missing(TEAM_FIELDING_DP);
FLAG_PSO_FIXED = missing(TEAM_PITCHING_SO);

if missing(TEAM_BATTING_SO) then TEAM_BATTING_SO = 736;
```

```
if missing(TEAM_BASERUN_SB) then TEAM_BASERUN_SB = 125;
if missing(TEAM_BASERUN_CS) then TEAM_BASERUN_CS = 53;
if missing(TEAM_FIELDING_DP) then TEAM_FIELDING_DP = 146;
if missing(TEAM_PITCHING_SO) then TEAM_PITCHING_SO = 818;
if missing(SB_PCT) then SB_PCT = 0.63;

if TEAM_FIELDING_E > 542 then TEAM_FIELDING_E = 542;
if TEAM_PITCHING_BB > 694 then TEAM_PITCHING_BB = 694;
if TEAM_PITCHING_H > 2059 then TEAM_PITCHING_H = 2059;
if TEAM_PITCHING_SO > 1095 then TEAM_PITCHING_SO = 1095;


if SB_PCT < 0.63 then SB_PCT = 0.63;
if TEAM_PITCHING_BB < 417 then TEAM_PITCHING_BB = 417;
if TEAM_PITCHING_HR < 25 then TEAM_PITCHING_HR = 25;
if TEAM_PITCHING_SO < 490 then TEAM_PITCHING_SO = 490;
if TEAM_BATTING_SO < 421 then TEAM_BATTING_SO = 421;
if TEAM_BASERUN_SB < 44 then TEAM_BASERUN_SB = 44;
if TEAM_BASERUN_CS < 30 then TEAM_BASERUN_CS = 30;
if TEAM_BATTING_3B < 27 then TEAM_BATTING_3B = 27;
if TEAM_BATTING_HR < 20 then TEAM_BATTING_HR = 20;
if TEAM_BATTING_BB < 363 then TEAM_BATTING_BB = 363;

TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B -
TEAM_BATTING_HR;


* The stepwise regression model with all variables selected since aic, bic and squared error were best of
the 3 models;

P_TARGET_WINS = 56.41909
                + 0.05178*TEAM_BATTING_1B
                + 0.0333*TEAM_BATTING_2B
                + 0.16274*TEAM_BATTING_3B
                + 0.10789*TEAM_BATTING_HR
                + 0.03136*TEAM_BATTING_BB
                + -0.02736*TEAM_BATTING_SO
                + 0.07572*TEAM_BASERUN_SB
                + -0.04694*TEAM_BASERUN_CS
                + -0.11258*TEAM_FIELDING_E
                + -0.11187*TEAM_FIELDING_DP
                + -0.01241*TEAM_PITCHING_H
                + 0.01353*TEAM_PITCHING_SO
                + -24.63666*SB_PCT
```

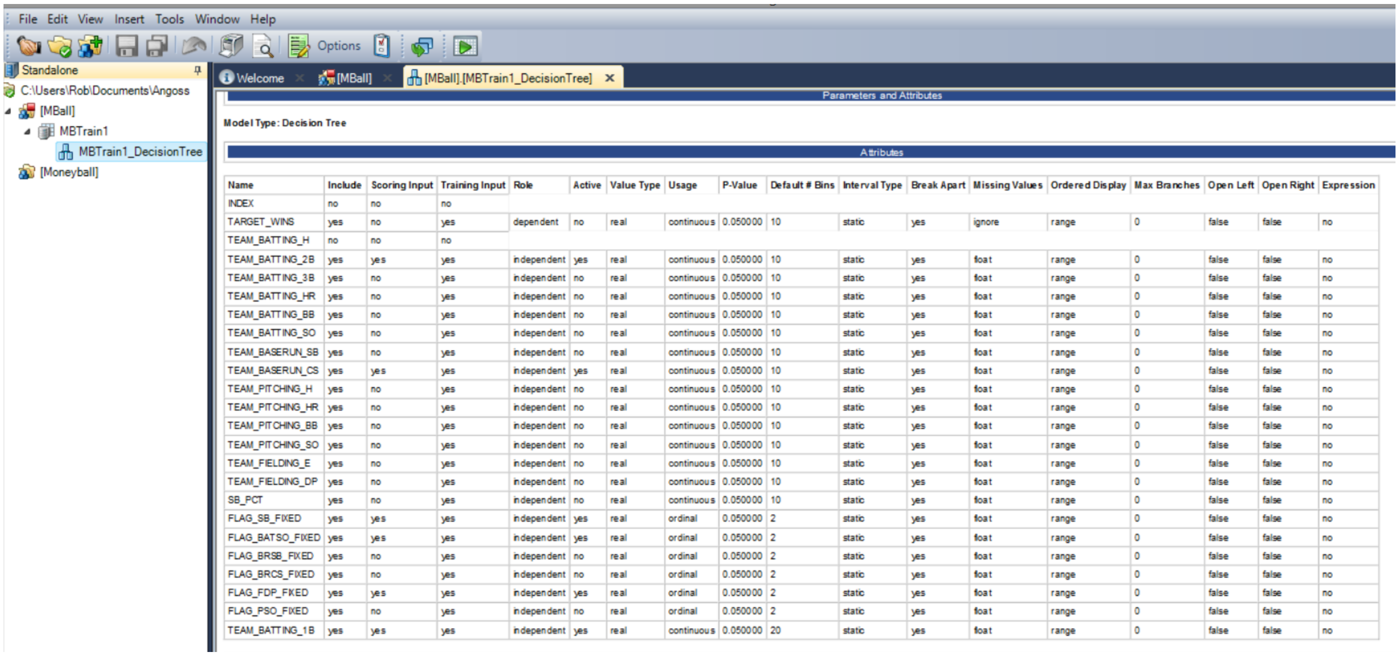
```
                              + 7.63914*FLAG_BATSO_FIXED
                              + 31.29647*FLAG_BRSB_FIXED
                              + 2.4961*FLAG_BRCS_FIXED
                              + 8.98023*FLAG_FDP_FIXED;

if P_TARGET_WINS < 50  then P_TARGET_WINS = 50;
if P_TARGET_WINS > 105 then P_TARGET_WINS = 105;

run;

* Verifies there are 259 results to match the number of obs in moneyball_test;

proc means data = moneyball_test_scores n;
TITLE "Number of obs in output file";
run;

* Keeps only desired columns;

data moneyball_results;
        set moneyball_test_scores;
        keep index PP_TARGET_WINS
run;


proc print data = moneyball_results;

TITLE "Moneyball Predictions";
un;

* Copies output to sas7dat file;

data MB.rtorp_pred411_s55__p_target_wins
        set moneyball_results;
run;
```

Bonus Bingo:

(20 Points) Use decision tree software such as Angoss for variable selection.

The variables listed as 'Active' in the screenshot were determined to be predictive in regards to TARGET_WINS. I implemented these variables in a PROC REG statement in SAS. The model did not perform well so I am not certain if I did something wrong.

Robert Torp Predict 411 Sec 55 Moneyball Assignment



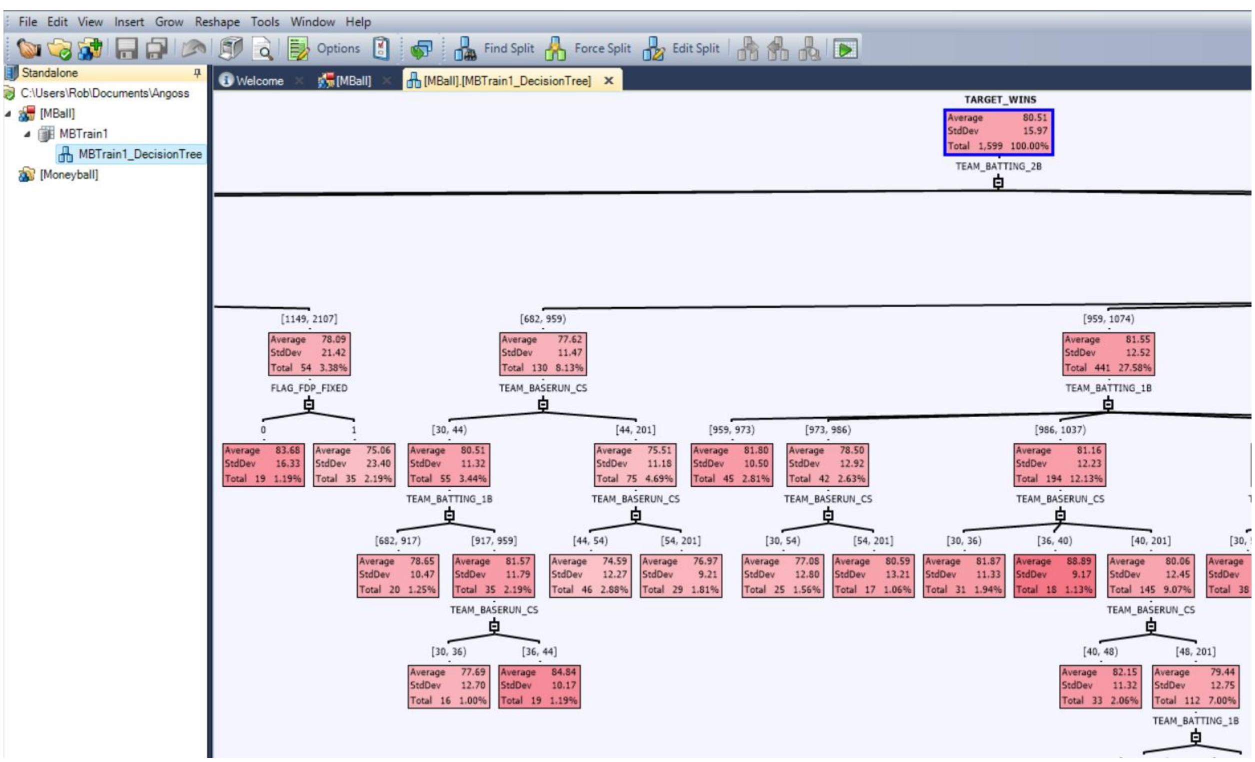
Model Type: Decision Tree

Parameters and Attributes

Attributes

Name	Include	Scoring Input	Training Input	Role	Active	Value Type	Usage	P-Value	Default # Bins	Interval Type	Break Apart	Missing Values	Ordered Display	Max Branches	Open Left	Open Right	Expression
INDEX	no	no	no														
TARGET_WINS	yes	no	yes	dependent	no	real	continuous	0.050000	10	static	yes	ignore	range	0	false	false	no
TEAM_BATTING_H	no	no	no														
TEAM_BATTING_2B	yes	yes	yes	independent	yes	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BATTING_3B	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BATTING_HR	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BATTING_BB	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BATTING_SO	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BASERUN_SB	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_BASERUN_CS	yes	yes	yes	independent	yes	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_PITCHING_H	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_PITCHING_HR	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_PITCHING_BB	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_PITCHING_SO	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_FIELDING_E	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
TEAM_FIELDING_DP	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
SB_PCT	yes	no	yes	independent	no	real	continuous	0.050000	10	static	yes	float	range	0	false	false	no
FLAG_SB_FIXED	yes	yes	yes	independent	yes	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
FLAG_BATSO_FIXED	yes	yes	yes	independent	yes	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
FLAG_BRSB_FIXED	yes	no	yes	independent	no	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
FLAG_BRCS_FIXED	yes	no	yes	independent	no	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
FLAG_FDP_FIXED	yes	yes	yes	independent	yes	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
FLAG_PSO_FIXED	yes	no	yes	independent	no	real	ordinal	0.050000	2	static	yes	float	range	0	false	false	no
TEAM_BATTING_1B	yes	yes	yes	independent	yes	real	continuous	0.050000	20	static	yes	float	range	0	false	false	no

A partial screenshot of the resulting tree in Angoss:



(10 Points) Use SAS Macros or use, in my opinion, good programming technique

Examples of SAS Macros in my code:

```
%let PATH =/folders/myfolders/sasuser.v94/411_data/Moneyball;
%let NAME = MB;
%let LIB = &NAME..;
libname &NAME. "&PATH.";
%let INFILE = &LIB.MONEYBALL;

*Explore response variable;

proc univariate data=&INFILE. plot;
	var TARGET_WINS;
run;

*Score Code Below;

%let results = MBValidate;

data score_file;
set &results.;
%let NAME = MB;
%let LIB = &NAME..;
libname &NAME. "&PATH.";
%let INFILE = &LIB.MONEYBALL_TEST;

data moneyball_test_scores;
set &INFILE.;
```

(10 Points) Hand in your SCORED FILE as a SAS DATA SET and save me to trouble of converting it.

I included the following code in my SCORED FILE to create a SAS DATA SET called rtorp_pred411_s55__p_target_wins. I then exported the file to turn in with the assignment.

```
* Copies output to sas7dat file;
data MB.rtorp_pred411_s55__p_target_wins
	set moneyball_results;
run;
```