Pitcher Performance in Major League Baseball

Advised by: Dr. Matthew Biesecker

Senior Capstone

Department of Mathematics and Statistics

South Dakota State University

Reece Arbogast

Fall 2023

Abstract

This paper uses independently collected data on pitchers in Major League Baseball (MLB) in order to analyze trends in their production and reliability. The goal of this paper is to use different predictors to create a logistic regression model that outputs the likelihood that a pitcher has increased performance and/or increased usage (in the amount innings pitched). These predictions will be based on the pitcher's statistics from the previous year. This paper investigates the logistic regression model and the function used to create it. Specific topics covered include the range of the function, parameters used in the function, and methods used for estimating these parameters. The paper will briefly investigate a model creation method known as StepAIC, as well as the influence of outlier data and collinearity. The effect that these have on the accuracy of the logistic model will also be discussed. The paper will finish by discussing results. This will include the accuracy of the model created, as well as comparing how accurate the logistic model is compared to other predictive models.

Pitcher Performance in Major League Baseball

1 Introduction

In this paper, we will be analyzing a set of Major League Baseball (MLB) pitching data to create a logistic regression model. Data has been collected on MLB pitchers from trusted sites such as baseballsavant.com[1] and baseballreference.com[3]. The data set includes 24 unique variables, ranging anywhere from first and last name to Fielding Independent Pitching (FIP) and Chase Rate Percentile. Table 1 gives the name and description of each variable included in the data.

Table 1: Variable Names and Descriptions

Variable Name	Description
First	First Name of Pitcher
Last	Last Name of Pitcher
Handedness	"Right" or "Left" – the Pitcher's throwing arm
Team	Team in 2022
Role	Role in 2022
Age	Age in years
Seasons	Seasons played in MLB
ERA2022	Earned Run Average in 2022 (Per 9 Innings)
ERAPlus2022	Earned Run Average Plus in 2022
IP2022	Innings Pitched in 2022
HitsPer9	Average Hits given up per 9 Innings
HRPer9	Average Home Runs given up per 9 Innings
SOtoBB	Ratio of Strike Outs to Walks
WHIP	(Walks given up + Hits given up) / Innings Pitched
FIP	Fielding Independent Pitching – Measures expected ERA
Velo2022	Fastball Velocity as League Percentile
SOPct	Strikeout % as League Percentile
HHPct	Hard Hit % as League Percentile (amount of hard hits a pitcher gives up)
WhiffPct	Swing and Miss % as League Percentile
ChaseRate	Swings induced out of the strike zone as League Percentile
ERAPlus2023	Earned Run Average Plus in 2023
ProIP2023	Projected Innings Pitched in 2023
BetterERAPlus	Measures whether or not a Pitcher improved in ERA Plus from 2022 to 2023
BetterIP	Measures whether or not a Pitcher threw more innings in 2023 than 2022

The goal of this paper is to use these pitching statistics to create a logistic regression model that predicts whether an MLB pitcher's ERA Plus (Earned Run Average Plus) will improve when outside statistics are inputted. ERA Plus as a statistic is a normalized version of earned run average (ERA). ERA is defined as the average number of earned runs a pitcher gives up per nine innings pitched. ERA Plus adjusts ERA so that the league average is 100, and each additional

point is a percentage above or below the league average. For example, an ERA Plus of 140 would be 40% above the league average.

2 Data Visualization

With the plethora of data available, many charts and models can be made to visualize the data. These types of graphs help to initially see the possible relationships between ERA Plus and some of the other variables. The figures below are some examples of relationships that can be discovered between ERA Plus and the other variables.

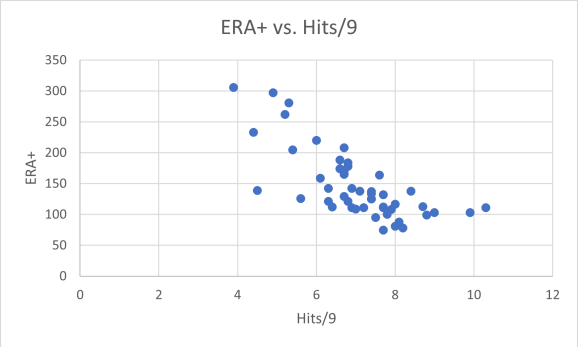

Figure 1: ERA Plus vs. HitsPer9 Graph (Microsoft Excel)

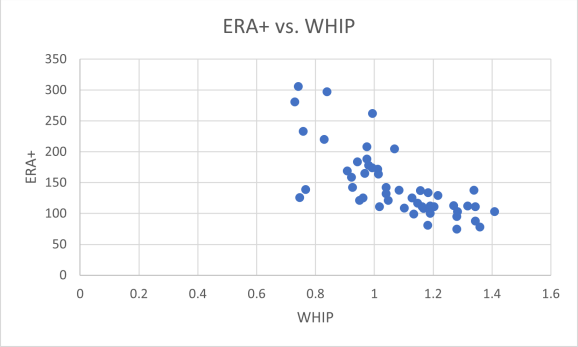

Figure 2: ERA Plus vs. WHIP (Microsoft Excel)

Both Figure 1 and Figure 2 display similar relationships. As HitsPer9 and WHIP decrease, ERA Plus generally goes up. However, it is not always as easy to spot these correlations and determine their significance.

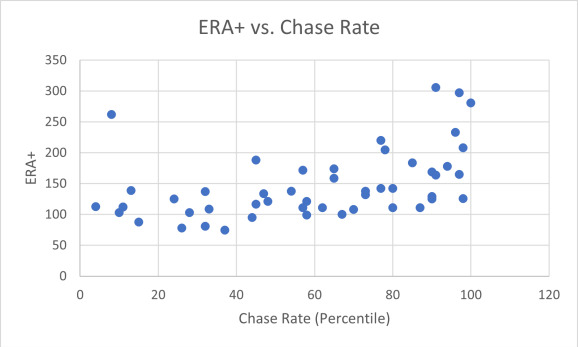

Figure 3: ERA Plus vs. Chase Rate Percentile (Microsoft Excel)

Figure 3 displays a much more vague relationship that occurs between Chase Rate and ERA Plus. There appears to be a positive correlation between the two variables, but with a slope of possibly negligible magnitude that may question statistical significance. The three models present different sorts of correlation, and it appears that singular variable linear regression can only do so much. Thus, the question arises - Can we use multiple variables to predict an increase or decrease in a specific pitcher's ERA Plus? In order to do this, we turn to Logistic Regression.

3 Logistic Regression

Logistic Regression is a technique that uses multiple predictor variables to determine the likelihood a certain event occurs. It is made up of variables of two classification types: exposure variables and control variables[2]. Exposure variables are dependent variables; they are what we are trying to predict. Control variables, then, are the independent variables used for this prediction. Additionally, the graph of the logistic model is shown in Figure 4.

Figure 4 shows in part why this particular model is so popular. The range of the graph is

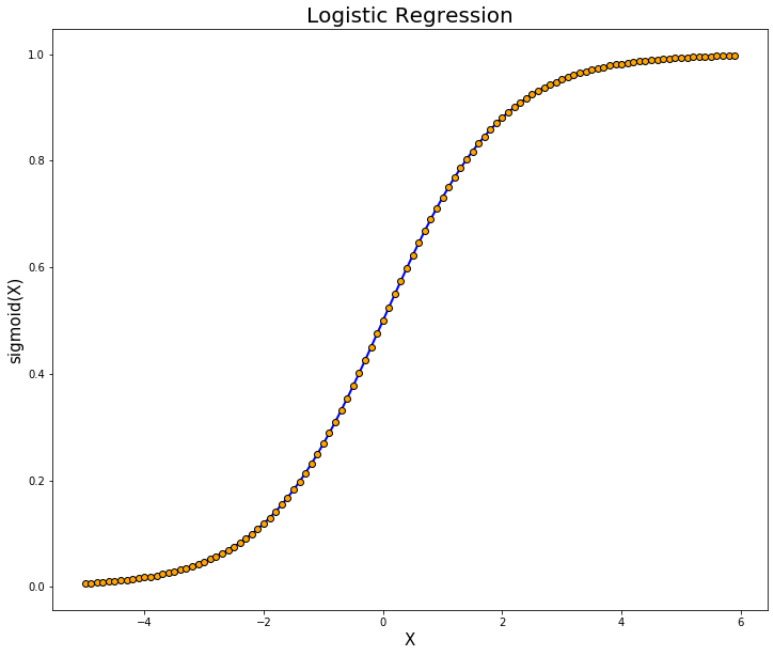

Figure 4: The Logistic Regression Model (Towards Data Science)

between 0 and 1, allowing its output to be interpreted as a percent likelihood. The logistic model follows the equation

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{1}$$

This will enable us to put a value on our variable z in accordance to our control variables. This model defines z as

$$z = \alpha + \beta_i X_1 + \beta_i X_2 + ... + \beta_k X_k, \tag{2}$$

where α and β are estimators based on our data and X_i represents each one of the control variables we are using. Equation 1 can be modified to the form

$$f(z) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i X_i)}}. \tag{3}$$

The key to solving our problem and producing an output probability lies in determining the estimators α and β_i[2]. To do this, statisticians commonly use the Maximum Likelihood Method, which is a way of estimating values of α and β_i that optimize the accuracy of the model.

It is important to discuss why the output of Logistic Regression is interpreted as a probability statement. To help understand this, first consider simple linear regression. The goal in linear

regression is to formulate a line that, when a predictor of value x is inserted, an expected value for y, the response variable, is outputted. Written in mathematical terms, basic linear regression with one predictor variable yields

$$E(y|x) = \alpha + \beta_1 x, \tag{4}$$

where α and β_1 are estimated parameters. For multiple predictors, this can be modified to the form

$$E(y|x) = \alpha + \sum \beta_i X_i, \tag{5}$$

which is the result commonly seen in multiple linear regression.

If the dependent variable is continuous, some type of linear regression appears to be a suitable method for prediction. However, the fact that our dependent variable is binary poses serious problems for a linear model. Note that one of the assumptions in using a linear model is that the dependent variable is roughly normal in distribution. With binary variables, that is simply not the case. Additionally, in order to solve the problem at hand, some value of probability or risk needs to be calculated. The very definition of our problem is based upon trying to calculate the probability of whether or not a particular pitcher improves, so a linear model may not be the best option[4].

Also, note that a linear model does not have any constraint on output values. In other words, the output of a linear regression model is dependent on the estimates of predictor coefficients[4]. For instance, for a line with positive slope, values of the dependent variable y increase without bound as x increases. This does not work well for our problem, as we are only concerned with predicted values between zero and one. Also, note that it is not necessarily logical for probability to increase at a linear rate for an entire range of predictor values, x. For example, consider COVID-19. Let one represent someone not getting COVID-19, and let zero represent someone contracting the virus. Let the predictor variable x be the number of booster shots a particular person gets in a six month span. If a person gets no booster shots, they would be more likely to contract the disease. If a person gets a booster shot, they would be more likely to not get the disease. However, is someone that gets five booster shots five times more likely to not contract

the disease than someone who only gets one? Arguably, no.

Thus, two major problems with the linear model have been identified. This type of model does not represent a probability statement, and probability is often non-linear. This is where the logistic model comes in. As seen in Equation 3, the logistic model modifies the expected value statement introduced in linear regression. This can be more easily seen in observing Figure 4. Notice how both of the problems identified with linear regression are resolved. The curve in Figure 4 is now non-linear and ranges from zero to one.

As to why this is the case, consider the denominator of $f(z)$ in Equation 3. It is true that, for positive values of x,

$$0 < e^{-x} < 1.$$

Then, the denominator of $f(z)$, $1 + e^{-(\alpha+\sum \beta_i X_i)}$, satisfies the inequality

$$1 < 1 + e^{-(\alpha+\sum \beta_i X_i)} < 2.$$

Therefore, since $f(z)$ has a positive numerator that is always less than a positive denominator, it is true that

$$0 < f(z) < 1.$$

Additionally, we discussed above that probability is often non-linear. Note that e^{-x} is an exponential curve, which is non-linear by nature. Thus, when basic function transformations are applied to it, as is done in Equation 3, the curve is still exponential. Therefore, we can verify that Equation 3, the logistic regression equation, represents a curve fitted for the nature of this problem.

We can also prove that the output of logistic regression is a transformation of a linear sum of the predictor variables (linear regression calculation). In order to do this, first consider something that is commonly related to probability statements - the odds[2]. Mathematically, odds,

denoted O_d, is defined as a function of probability, where

$$0_d = \frac{P(X)}{1 - P(X)}$$

For our proof, we will use what is called the logit function of $P(X)$, where the logit function is the natural log of the odds. This is why this function is also know as the "log odds" function[2]. Therefore,

$$\begin{aligned} logitP(X) &= ln\left(\frac{P(X)}{1 - P(X)}\right) \\ &= ln\left(\frac{\frac{1}{1+e^{-(\alpha+\sum \beta_i X_i)}}}{1 - \frac{1}{1+e^{-(\alpha+\sum \beta_i X_i)}}}\right). \end{aligned} \tag{6}$$

We can transform the denominator, $1 - P(X)$ so that

$$\begin{aligned} 1 - P(X) &= 1 - \frac{1}{1 + e^{-(\alpha+\sum \beta_i X_i)}} \\ &= \frac{e^{-(\alpha+\sum \beta_i X_i)}}{1 + e^{-(\alpha+\sum \beta_i X_i)}}. \end{aligned}$$

Substituting this back into the logit function, we get that

$$\begin{aligned} logitP(X) &= ln\left(\frac{\frac{1}{1+e^{-(\alpha+\sum \beta_i X_i)}}}{\frac{e^{-(\alpha+\sum \beta_i X_i)}}{1+e^{-(\alpha+\sum \beta_i X_i)}}}\right) \\ &= ln(e^{\alpha+\sum \beta_i X_i}) \\ &= \alpha + \sum \beta_i X_i, \end{aligned} \tag{7}$$

which is the resulting equation from linear regression with multiple predictor variables[2]. This is significant when the output of simple linear regression is considered. In linear regression, the output line, also known as the line of best fit, minimizes the error between the line and each data point. Thus, with this proof, we have shown how error is minimized in logistic regression. We simply take the best fit line outputted by linear regression and apply basic function transformations. Now, there exists a line with minimum error that still fits the context of our problem.

4 The Logistic Model

The whole goal of this experiment is to determine the likelihood that a pitcher will see improvements in effectiveness and usage based on a previous season's statistics. In order to do this, we create what is called the Logistic Model. The Logistic Model takes a certain number of independent variables for a specific subject and combines them in an optimal way to give the best possible prediction on the exposure variable. For our purposes, the independent variables are the statistics collected, such as FIP, ERA Plus in 2022, hits given up per nine innings, etc. The subjects, obviously, are each different pitcher, and our exposure variables (what we are trying to predict) are ERA Plus in 2023 and Projected Innings Pitched in 2023.

In order to make accurate predictions, we need test data for which we have determined whether or not a pitcher has gotten better and has been used more. This is the purpose of binary variables $BetterERAPlus$ and $BetterIP$. These variables determine whether or not a certain pitcher did or did not see improvement in each category. For example, Yankees' right-hander Gerrit Cole saw improvement in the 2023 season. His improved ERA Plus from 2022 to 2023 is denoted with a 1. Sandy Alcantara, who was much worse in 2023 than in 2022, had a lower ERA Plus, so his decreased production is denoted with a 0. This same logic is also applied to the $BetterIP$ variable category.

Because our model's output is determined as a probability statement, having these 1's and 0's in our test data is extremely important. This allows the model to interpret relevant predictors in context with the questions we are asking. These binary variables show whether or not each pitcher "succeeded" or "failed" in being more productive or pitching more innings. Once again, the key factor in interpreting our model is understanding that it is a probability statement. Thus, Equation 3 can be transformed to

$$P(X) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i X_i)}},$$

where $P(X)$ represents that probability of Pitcher X having a higher ERAPlus or more innings pitched over the course of the season.

Additionally, an important aspect in creating a model is to prevent overfitting[4]. To combat this, we will test our variables for multicollinearity. This condition occurs when a predictor variable (or multiple predictor variables) can be roughly determined by one or more other predictors. For example, in our data set, seasons played and a player's age may be collinear, as they both have influence on the amount of time a pitcher has spent in the league.

Additionally, significant outliers in test data may greatly affect our model and its prediction accuracy[2]. Outliers in one or more independent variables can greatly affect coefficients for regression for the overall model. If this happens to be the case, a higher likelihood for (and perhaps magnitude of) error can be introduced into the model. For example, in our data, Tampa Bay Rays right-hander Jeffrey Springs has an ERA Plus in 2023 of 760. This is over 500 units higher than his next closest counterpart. This drastic outlier can greatly skew regression coefficients and affect the accuracy of our model, so it will be removed.

5 Analysis

In the first steps of model creation, the data is subset into "training" and "testing" sets. The training set is used to create the model. Thus, the initial model (and all model adaptations afterwards) are created from analyzing the training data. The testing set is used to simulate outside data. The test set contains the true values of whether or not a certain pitcher improved or not, and since the test set had no influence on the model, we use it to check the accuracy of the model's predictions. For our purposes, we will utilize a 70/30 split. Thus, 70% of the data is partitioned into the training set, and the remaining 30% is used as the testing set.

To start, a logistic model is created using all predictor variables. The purpose of this is to simply set a baseline. This allows us to check initial diagnostics and make corrections. When analyzed, the initial model produced very poor results. This is to be expected. As discussed previously, multicollinearity poses a very real threat to the accuracy of a logistic model. After using a VIF test for multicollinearity, we found that nearly every variable had collinearity concerns. A VIF test essentially measures correlation between independent variables and assigns it

a score based on how well it can be explained by a combination of other independent variables. The results from the intial VIF test can be seen in Table 2.

Table 2: Initial VIF Scores

Variable Name	**VIF Score**
Age	8.59
Seasons	8.87
ERA2022	17.49
ERAPlus2022	8.79
IP2022	1.57
HitsPer9	39.93
HRPer9	22.30
SOtoBB	12.84
WHIP	40.67
FIP	37.03
Velo2022	2.39
SOPct	25.65
HHPct	2.34
WhiffPct	8.45
ChaseRate	4.44

Conditions of a VIF test state that any VIF score greater than roughly five requires investigation for collinearity. As seen, almost every variable has a score well over five, proving that multicollinearity in our model is a big issue. However, this is to be expected. Consider how baseball statistics are defined. Often times, they are calculated using a combination of other stats. For example, WHIP is calculated from the formula

$$WHIP = \frac{Walks + Hits}{InningsPitched}.$$

Thus, we would expect that WHIP could be explained by independent variables such as HitsPer9 and Innings Pitched. Because of this, multicollinearity could be a huge problem just because of the nature of our data.

Additionally, the first model showed that almost none of its independent variables were statistically significant. If a variable is not statistically significant, we can say that its inclusion in the model is necessary and important. With statistically insignificant variables, there is a possibility that our model gets "lucky" and produces results that cannot be replicated with consistency. The fact that our model had almost no statistically significant variables suggests that we can not trust the output of this model. Only the variables SOtoBB and SOPct had p-values less than $\alpha = 0.05$ in the initial logistic model's diagnostics. These facts suggest that overfitting is an issue. Again, this is to be expected, as every possible predictor variable was used in creating this model. Thus, in future adaptations of the model, our goal is to eliminate both overfitting and multicollinearity.

The next step in creating a final logistic model is using StepAIC. StepAIC is a function in the programming software R that performs backwards selection to create a model with minimal predictor variables. The exact methods behind StepAIC are outside the scope of this paper, but the goal in using this method is to create a model using less predictor variables in order to prevent overfitting and multicollinearity. However, this process of formulating an accurate model causes a few issues. StepAIC consistently produces a model based upon four predictor variables. However, when it was rerun and different observations were randomly selected into the training and testing data splits, the predictor variables used were not always the same. Additionally, collinearity was not completely eradicated. Each time a StepAIC model was calculated, at least one of the optimal predictor variables selected had a VIF score greater than five. With collinearity still being a frequent issue, we want to ensure that it is not present in our next updated model.

One way of eliminating covariance between predictor variables is dimension reduction. Dimension reduction is a technique used in data analysis that decreases the amount of predictor variables used in model creation. This helps our analysis in two ways. First, with a lesser number of predictors in our model, overfitting should not be an issue. Also, because we reduce the dimension of the data, we expect collinearity between variables to be lessened. For example, we discussed

earlier the likely collinearity between WHIP, HitsPer9, and Innings Pitched. Dimension reduction performs operations that either removes two of the three variables, or it creates a single variable from a linear combination of the three collinear variables.

The form of dimension reduction we will use is known as Principal Component Analysis (PCA). PCA calculates what is known as the "principal components" of the data set. Once again, the mathematics behind PCA is outside of the scope of this paper, but there are a few important characteristics of the principal components that we will use in analysis of our model. First, for x predictor variables, we will have x principal components. These components are ranked in order by how much of the variance in the data set they account for. For example, for three predictor variables, it might be true that the first principal component accounts for 74% of the variance, the second may account for 20%, and the third 6%. Note that the sum of the variances accounted for by each principal component will always sum to one. Additionally, it can be shown that each principal component is independent from all other principal components. Thus, the covariance between each principal component is zero, and we will have completely removed any collinearity in our predictors.

To create our final model, PCA is performed on our twenty-two numeric predictor variables. The scree plot in Figure 5 displays the percentage of variance accounted for by each principal component.

In creating our model, we want to find the "elbow" of the scree plot. This is the point where the variance added by each additional principal component becomes minimal. For our purposes, we will make the cutoff at principal component three. In this case, the first three principal components account for roughly 62% of the total variance in the data.

Next, a logistic regression model is calculated using the first three principal components of the data on the training set. The choice of using these three components allows us to minimize overfitting while still accounting for the majority of the variation in the data. The true test to see if these principal components are enough to satisfy a sufficiently accurate model will be when the model is tested on both the training and testing data sets.

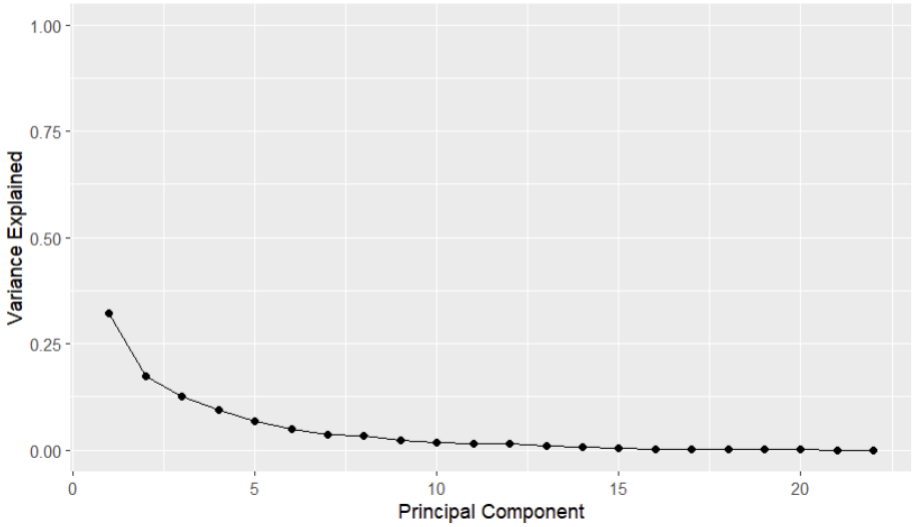

Figure 5: Scree Plot of Variance Explained (RStudio)

6 Results

The model is now tested on both training and testing data sets. We will discuss why this is the case later on. Note that a success, in this case, is defined as the number of true positives and true negatives in our predictions. For example, a true positive is defined as the model predicting that a pitcher will have an increased ERA Plus, and that pitcher actually does achieve a better ERA Plus. Similarly, a true negative is when the model predicts a worse ERA Plus, and that pitcher actually has a worse ERA Plus. Essentially, the number of true positives and true negatives is the number of correct predictions made by the model. Table 3 displays the percent accuracy of the model's predictions over ten separate trials.

We use multiple trials in order to expose the model to different randomly selected training and testing data. In doing so, we simulate what happens when the model is exposed to different outside data. This further validates the model by giving us an average percent accuracy to analyze. Note that, on the training data, our model predicts BetterERAPlus with 97.18% accuracy. Such a high percent accuracy indicates that the model is very reliable on the training data. This is to be expected because we used this data to create our model. The true test is the model's accuracy on the testing data, i.e., data it has not been previously exposed to. As seen in Table 3, the model predicts the testing data with 93.05% average accuracy. Again, this mark is rel-

Table 3: PCA Model Percent Accuracy

Trial	**ERA Plus Training Accuracy**	**ERA Plus Testing Accuracy**	**BetterIP Training Accuracy**	**BetterIP Testing Accuracy**
1	100	96.4	91.4	82.1
2	98.6	92.3	84.7	80.7
3	100	96.4	90.0	78.6
4	96.7	86.8	80.0	92.1
5	97.0	87.1	82.1	90.3
6	100	97.3	93.3	78.9
7	96.7	88.9	90.3	80.5
8	94.5	96.0	87.7	76.0
9	95.7	89.3	87.7	78.6
10	92.6	100	85.3	76.7
Avg.	97.18	**93.05**	87.48	**81.45**

atively high. We believe that, if a random pitcher's statistics are inputted into the model, our model's prediction on whether or not he will improve will be correct with roughly 93% accuracy. Similarly, Table 3 displays that we predict BetterIP with 87.48% accuracy on the training data. In comparison, the model predicts BetterIP with 81.45% accuracy on the testing data. While the BetterIP model does not perform quite as well as the BetterERAPlus model, it still does predict correctly roughly eight out of ten times. In Major League Baseball, we often see certain anomalies, and consistency is hard to come by, as teams are constantly adapting to exploit an opposing players weaknesses. Thus, being able to correctly predict a player's success even eight out of ten times can prove incredibly valuable to any organization.

Lastly, it is important to discuss whether or not we trust our results. Remember, the goal in creating a PCA model is to eliminate both overfitting and collinearity. Consider that, if a model is overfit, it will perform significantly better when predicting the training data in comparison to the testing data. The reasoning for this relates back to linear regression. For example, if

we have 20 predictor variables and 20 data points we are trying to fit a line to, it is possible to formulate a line that will have no error from any of the data points. However, when this model is exposed to outside data, we expect there to be error, as it is incredibly unlikely that each new data point will fall exactly on the regression line. Note that our models predict the training data extremely well, but they also predict the testing data with almost equal accuracy. Also, take into account trial eight in Table 3. Note that in this trial, the BetterERAPlus model actually performed better when predicting the testing data in comparison to the training data. Similarly, this occurs in trial five for the BetterIP model. Based on these reasosn, it is safe to assume that overfitting is not an issue in both models.

Consider the issue of multicollinearity. To say that two predictors are collinear is equivalent to saying that they are correlated. We stated earlier that in using PCA, each principal component will be independent from each other and have no correlation. Therefore, we conclude that our model crushes collinearity and prevents overfitting. Thus, the prediction accuracy can be trusted.

7 Conclusion

Logistic regression provides a way to further analyze and estimate outcomes of binary variables. Throughout the process of data manipulation and model creation, we found that collinearity and overfitting cause issues in the reliability of a logistic regression model's output, especially in the context of baseball statistics. However, when these two issues were accounted for, logistic regression was able to provide a successful estimate for the likelihood of a pitcher's improvement.

For further analysis, one can consider other classification techniques, such as Support Vector Machines and Random Forest Models. The accuracy of these models can be compared with logistic regression, and a decision can be made on which model fits the data best. Additionally, other techniques of data transformation can be used to account for problems like collinearity and overfitting. Models created from these adjacent methods can also be compared to the logistic regression model.

Table 4: Comparing ERA Plus Between Models

Method	**Training Accuracy (%)**	**Testing Accuracy (%)**
Logistic Regression	97.18	93.05
Support Vector Machines	98.58	91.61
Random Forest	99.53	81.68

Table 5: Comparing BetterIP Between Models

Method	**Training Accuracy (%)**	**Testing Accuracy (%)**
Logistic Regression	87.48	81.45
Support Vector Machines	95.89	79.95
Random Forest	99.13	79.89

Tables 4 and 5 compare the logistic regression model's percent accuracy to other well-known classification methods. We can conclude that, for this example, logistic regression is the best choice based on it's average percent accuracy on the test data. The results suggest that both the Support Vector Machine and the Random Forest are overfit, and logistic regression still outperforms both models on test data. In addition to these models, logistic regression can also be tested against classification methods such as neural networks or K-nearest neighbors. The training and testing data can also be split differently and compared. For example, we can compare the results obtained in our 70/30 split to an 80/20 or 60/40 split. Small adjustments such as these can all attribute to obtaining an even more accurate and precise model.

References

[1] "Baseball Savant: Trending MLB Players, Statcast and Visualizations." *Baseballsavant.com*, 2023, baseballsavant.mlb.com/.

[2] Kleinbaum, David G., and Mitchel Klein. *Logistic Regression: A Self-Learning Text*. Springer Science + Business Media, 2011.

[3] "MLB Stats, Scores, History, Records." *Baseball Reference*, 2023, www.baseball-reference.com.

[4] Vittinghoff, Eric, et al. *Linear, Logistic, Survival, and Repeated Measures Models*. Springer Science + Business Media, 2004.

Biography

Reece Arbogast is a Mathematics and Statistics student at South Dakota State University. He is originally from Sioux Falls, SD. Reece plans to graduate in the spring of 2024 and to further pursue a Master's Degree in Statistics. As a collegiate baseball player, Reece hopes to someday be a performance analyst in Major League Baseball.