Stop The Bleeding: When Do We Call The Pen?

CIS 5930: Data Mining Proposal

Jason Hawkins, Michael Woodham
https://github.com/Maw1395/Baseball

I. Introduction

Here's the situation: Your pitcher is Tim Lincecum of the San Francisco Giants. It's the year 2008 and you're in game number 63 on the season for the Giants, playing the Washington Nationals. Washington hasn't fared too well this season, and they're coming into San Francisco with a record of 25-37, they're 11.5 games back on the Phillies. Granted, it's only June, but it looks like it's going to be a long season for the Nationals. The Giants on the other hand, are firmly in the playoff race, riding on the backs of their elite pitching core. The 7th inning has just concluded. Lincecum has thrown 83 pitches, faced 25 batters, has let the Nationals score only 1 run. Though, it's a long season, and there's a lot of games still to play. Should you take him out of the game now, and give it up to your bull-pen or let it ride and coast to an easy victory with one of the best pitchers in the MLB on the mound?

Baseball has been America's pastime for over one-hundred and fifty years. One of the most controversial topics is how long to keep a pitcher in a certain game for. It starts at as young as little league (ages 10-13) where the amount of pitches dictates how many days of mandatory rest a pitcher needs after. This has become a rather contentious topic within the circles of little league baseball, but it's application to the longevity of pitchers in their later years has been fairly well reported.[1] This continues through all levels of baseball.

The general idea of this paper has changed as the project has gone along. Initially, we wanted to make a meta analysis of when managers should go to the pen, this proved to be the incorrect question to ask given the dataset that we acquired. The idea changed from when they should, to when they do go to the pen and what factors lead to this decision. Data aggregated includes runs given up by the pitcher, total number of pitches thrown, strikes, balls, opposing slugging percentage, etc. All of these data points will be covered throughout the paper. What we found was that longevity of pitchers in a majority of cases, especially at the major league level, is more of a factor of time and health concerns for the pitchers. Our model looks for outliers within these norms, and usually when these are prevalent, it recommends an immediate pull of the pitcher.

II. Literature Survey

There has been limited research in the field of data mining as it pertains to baseball, such as *Forecasting MLB World Champions Using Data Mining,* authored by Robert "Edward" Egros from Northwestern University. With this there have been many University level projects that are posted online, but none of which address the problem we present. Most of the research that has been conducted deals with the prediction of games won vs games lost. Though, most of the methods used to mine data could prove useful for our attempts to gain information to base our model off of.

One big inspiration we did have for this project was *A Data-driven Method for In-game Decisions Making in MLB* by Gartheeban Ganeshapliiai, and John Guttag. In their paper they build a model that potentially could, "lead to better on-field decisions by predicting a pitcher's performance in the next inning."[2] This is similar to what we want to accomplish, but instead of predicting future innings, we go out-by-out to determine the optimal time to pull a pitcher.

III. Methodology

The problem we're trying to answer boils down to a binary classification problem. Our data-set is broken up by each out in a baseball game, with a multi-feature vector of length eighteen. We then label each row of our data-set with a zero or one. With zero being if the

[1] Erickson B, et al, Exceeding Pitch Count Recommendations in Little League Baseball Increases the Chance of Requiring Tommy John Surgery as a Professional Baseball Pitcher, The Orthopaedic Journal of Sports Medicine, 5(3), 2017

[2] Ganeshapillai, Gartheeban, and John Guttag. "A Data-driven Method for In-game Decision Making in MLB." *Sport Analytics Conference*. 2014

manager subbed the pitcher out of the game after the current out, or a one otherwise. Based on this knowledge we decided to use the random forest algorithm for this binary classification problem. The reasoning behind this is that decisions trees (or random forest which is basically a multitude of decisions trees) excel at classifying datasets that have a multi-feature vector into a binary result.

The basic idea behind a Random Forest classifier is creating an arbitrary n amount of decision tree models, and taking a majority vote to classify a sample. In data-mining terms, random forest creates an ensemble of Decision Trees, where a majority of time is “trained with the ‘bagging’ method”.[3] Figure 3.1 shows a visual representation of how this is done.

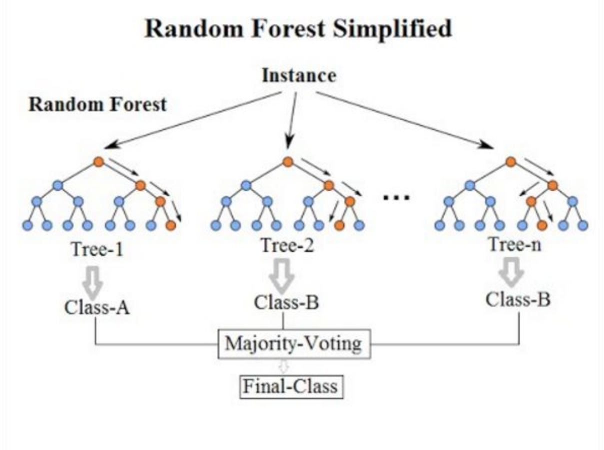


Figure 3.1

The decision trees in the Random Forest were trained on a testing set in which each sample contained eighteen attributes. The eighteen chosen are believed by us to have the most weight on when to pull a starting pitcher. Table 3.1 lists all the eighteen attributes that we had chosen.

Attribute	**Definition**
Strike Count	Aggregated number of strikes a pitcher has thrown
Ball Count	Aggregated number of balls a pitcher has thrown
Pitch Count	Total number of pitches thrown
Strike to Ball ratio	Number of strikes divided by number of balls thrown
Outs	Number of current outs
Inning	Current inning
Slugging Current	Slugging percentage of the current batter
OBP Current	On base percentage of the current batter
OPS Current	On base percentage plus slugging of the current batter
Slugging Average	Slugging percentage average of the current batting team
OBP Average	On base percentage average of the current batting team
OPS Average	On base percentage plus slugging average of the current batting team
Runs	Number of runs given up by the pitcher
Hits	Number of hits given up by the pitcher
Walks	Number of walks by the current pitcher
Strikeouts	Number of strikeouts by the current pitcher
Home Runs	Home Runs conceded by the current pitcher
On-base	Current positions of on-base runners

Table 3.1

[3] Donges, Niklas. "The Random Forest Algorithm – Towards Data Science." Towards Data Science. February 22, 2018. Accessed December 09, 2018. https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

IV. Stats Relevant To A Game and Season

What is a pitch?

A pitch is defined as a ball thrown to a batter where the batter has the opportunity to either swing or not swing at the ball

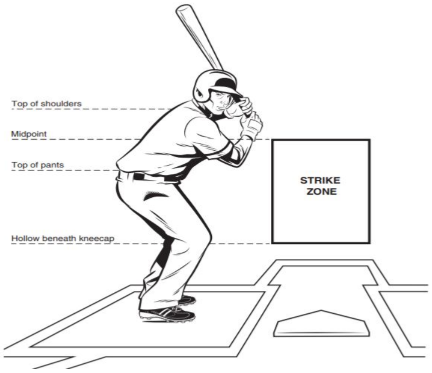
Average Number of Pitches Thrown In A Game : 96.6

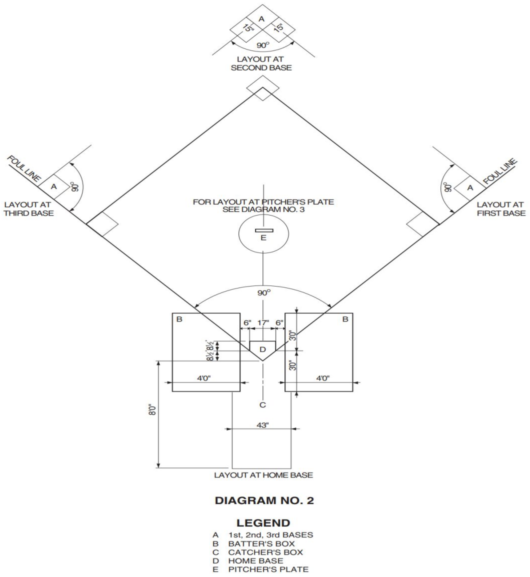
Why would a batter not swing at a pitch?

A batter would not swing at a given pitch if he thinks that it is out of his strike-zone. The strike-zone's y axis is defined to be from the top of the batter's shoulders to the batters knees, and it's x-axis being the width of the home plate (17"). [4]

Average Number of Balls: 31.1

Average Number of Strikes:65.5





What is a strike?

A strike can be accumulated in 7 possible ways, but here we will only cover the two most common.

1. The hitter does not make contact with a ball as it passes through his strike zone
2. The hitter makes contact with a ball whether is be in his strike zone or not, and it is sent out of the field of play, eg a *foul ball.*

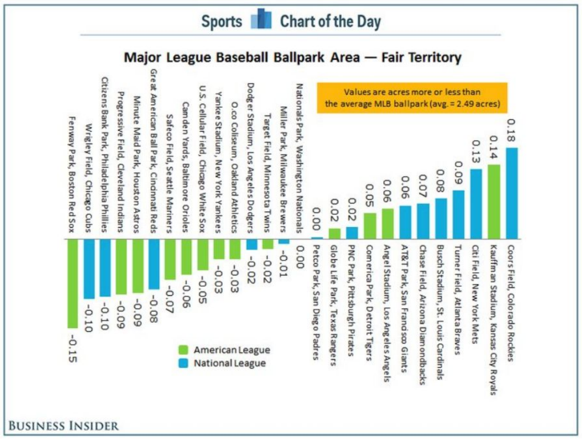
What is a ball?

A ball is a pitch which the batter chose to not swing at, and was outside of the batter's strike zone.

What is the field of Play?

-90° and +90° from the plate to the back wall, the back wall is not at a predefined distance, it can vary by field. For example, Boston's Fenway park hosts the smallest field of play in baseball, while the Colorado Rockies host the largest field of play, see chart below.[5]

90 feet at +90° to first base, 90 feet at -90° to third base. If a ball touches the ground in this field, then proceeds to go out of the 90° bounds past this point, eg further than 90 feet at + or - 90°, the ball is considered *fair* else if it never makes contact with the field of play or makes contact with the field of play and goes out of bounds prior to crossing the 90 feet threshold, it is considered *foul* and counts as a strike. If the ball stays within the bounds and never goes out, it is also considered *fair*. If the ball goes over the back wall within the + or -90° threshold, it is considered to be a *homerun*, and all players occupying a base, including the batter score.



[4] Official Baseball Rules, MLB, p158 & 155, 2018

[5] Gaines, C, Business Insider, MLB Ballpark Sizes Show The Immense Difference Between Fenway Park And Coors Field, 2014

What is the count?
The count is defined in the following notation b-s where b is the number of balls and s is the number of strikes. If the pitcher throws 4 balls, then the batter advances to first base, he can not stay at home plate, he must advance. This is considered to be a walk. If the pitcher throws 3 strikes, then the batter is considered to be out. If the number of strikes equals 2 and the batter hits a foul ball, then the number of strikes stays at 2. In order for a batter to be considered out at the plate, the pitcher must throw a strike in which the batter makes no contact with.

How is an out recorded?
An out can be recorded in a multiple number of ways, here we will only address the 4 most common:
1. Three strikes are recorded
2. The batter makes contact with the ball and it is caught by a fielder before it hits the ground
3. A ball is thrown to a base before the baserunner makes it to the base and it is a *force out*
Force out:

> Once the batter places the ball in the field of play he must advance to first base. If the runner does not make it to first base before the ball, he is considered to be out
>
> Two runners cannot occupy the same base at the same time, as such it is alway the responsibility of the front-runner to advance, if the ball reaches his base destination, before him he is considered to be out

4. A runner is touched with a ball before he makes it to a base. The ball cannot be thrown at the runner, a fielder must first catch it and apply a tag to a batter.

How is a runner considered *Safe?*
The runner is considered safe if he occupies a base before he is *tagged* or *forced out*. The runner can then stay on that base and be considered safe before he is forced to move, or chooses to advance at his own course. This can happen at any point that the ball is in play.

When is the ball in play?
The ball is in play at all times, unless a time out is called for any particular reason. The number of time-outs is not limited and are granted at the umpires discretion. A baserunner, batter, or fielder may call time at anypoint unless a play is in progres, e.g. the ball is not in the hands of a fielder within the infield.

How is a run Scored?
A run is scored once a player has made contact with all 4 bases without getting out. This can be done by a hit and running to all 4 bases, or a homerun where the batter is given all 4 bases. He must still make contact with all 4 bases, even on a homerun

Average Number of Runs Given Up: 2.7

What is an inning
An inning contains two parts, the top of the inning and the bottom of the inning. At the top of the inning, the home team is pitching and on defense, at the bottom of the inning the away team is pitching and playing defense. A inning is concluded when 3 outs are recorded. Inning notation is defined as 'I'.'O' where I is the number of innings and O is the number of outs. A game concludes when 9 innings have been pitched, and a team is ahead, or if the game is tied after 9 innings, until an inning has concluded and one team is ahead of another. As such the longest game in baseball history was 33 innings.

Average Number of Innings Pitched: 6.3

What is Slugging Percentage?
Slugging percentage is (Total Bases / At Bats). Total Bases being defined as how far the batter made it after immediately making contact with the ball, not how far the batter advanced overall. At Bats excludes walks.

What is On Base Percentage?
On Base Percentage is (the number of times the batter made it on base by any means be it a walk or hit / Total Number of At Bats). At Bats includes walks.

V. Description of Implementation

The first objective for us was to collect relevant data for experimentation. Since there were no publicly available databases with the information we desired, we decided to create our own web scraper. We implemented this using python 2.7, with get from the requests library, and

BeatifulSoup and Comment from the bs4 library. The website we aggregated our data from was: baseballreference.com. The python script imported the information gathered from baseball reference into a SQLite database.

Next, we created more scripts in python that dealt with the following: converting the SQLite database into a .csv (comma separated values) file, converting the newly acquired .csv into data samples with desired attributes, correctly labeling the samples, and a testing script that built a classification model. The reasoning behind converting to .csv was that it made it easier to work with the data in python with that format. For Classification modeling and testing we used the python library sklearn, and we used Panda dataframes to store the data once inside of python.

For the classification we implemented many different models: SVM, K-Nearest Neighbors, Decision Tree, and Random Forest. From the four classifiers used, Random Forest came back with the best results. For the Random Forest classifier we decided to go mostly with the default options provided by the sklearn library. For instance we used the Gini impurity to measure the quality of a split, the max depth of the tree is equal to none (meaning the tree expands until all leaves are pure), and the minimum samples required to split an internal node is equal to two. The one option we did explicitly define was the number of estimators, or the number of trees in the forest. The final testing we set this value to be equal to one-hundred.

V. Data We Acquired And Why

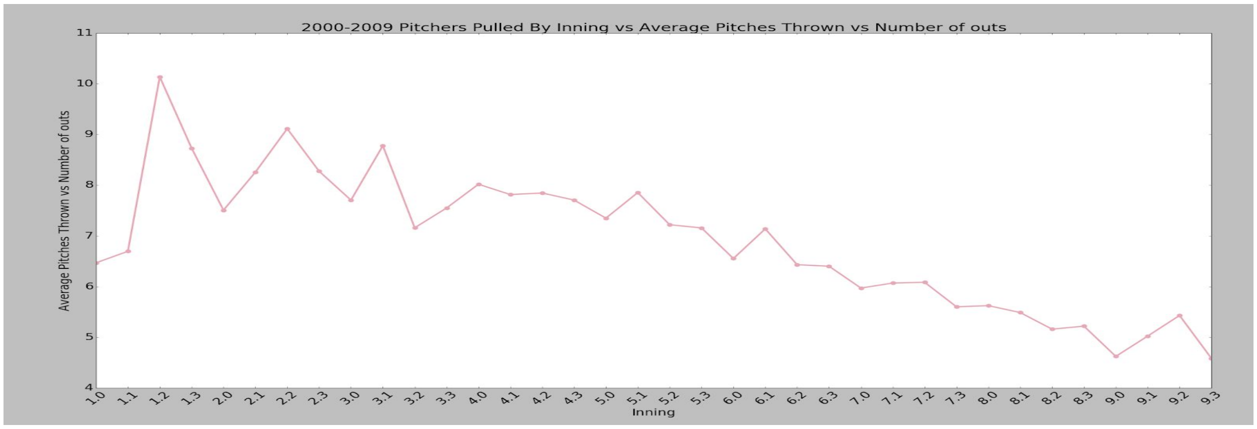
Strike Count: Contributes to Pitcher Accuracy
Ball Count: Detracts From Pitcher Accuracy
Walk Count: Detracts From Efficiency
Strikeout Count: Contributes to Efficiency
Strike-Ball Ratio: Gives us an Efficiency Score
Home-Runs: Gives us a *Death Clock*
Pitch Count: Contributes to Endurance
Run Count: Contributes to Efficiency
Outs and Inning: Gives us a Time Table
Slugging, ops, and obp: Gives us information on the quality of batter
Average Slugging, ops, and obp: Gives us information on what the pitcher has faced throughout the game, and how the current batter differs from these notions
On-base: Gives us the Current Scenario

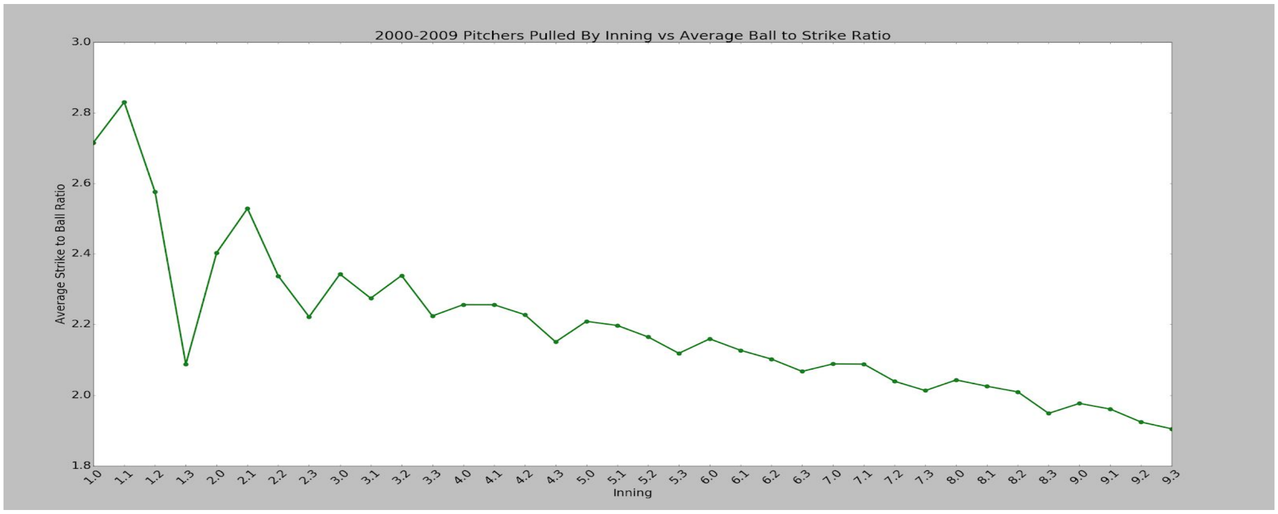
VI. Experimental Overview

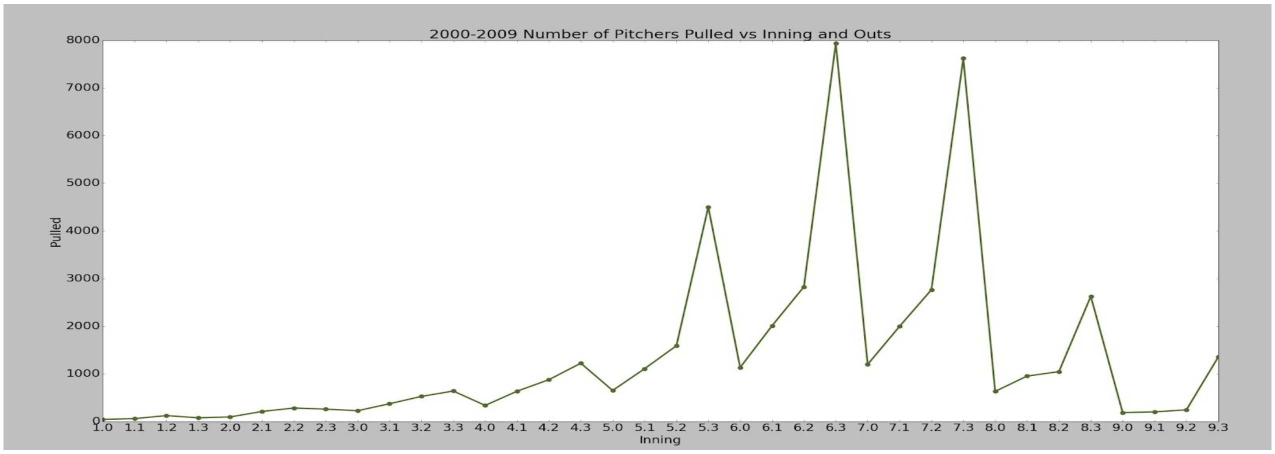
For our experiments we aggregated 1,271,828 data samples from baseball reference. Though, after purging the data of inconsistent data samples our final sample size was 1,228,074. Our experiments consisted of testing four different classifiers as described before: SVM, K-Nearest Neighbors, Decision Tree, and Random Forest. The SVM classifier we could never get to converge even with increasing the max iterations of the solver. For the next three classifiers we were able to get tangible results.

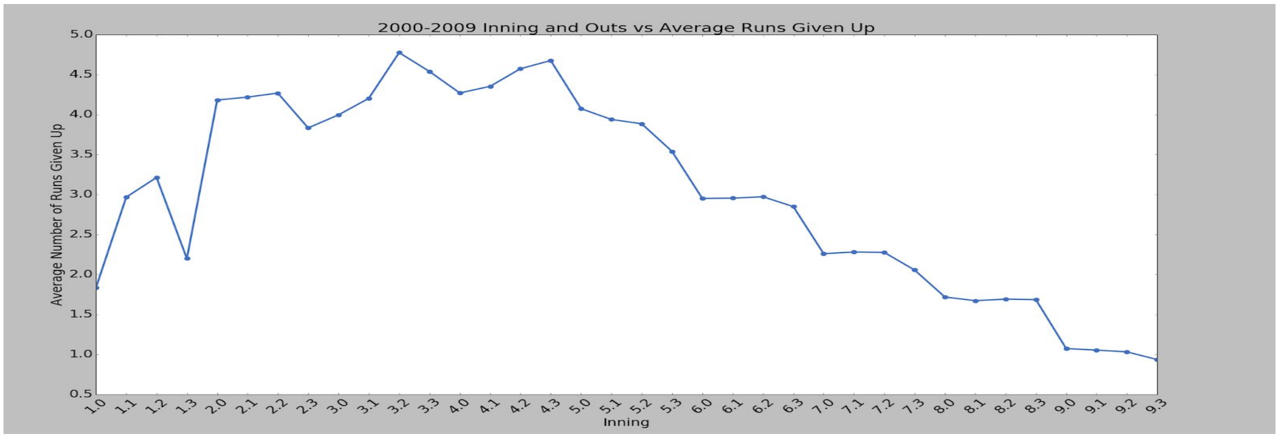
VII. Overall Statistical Graphical Representation

The 4 Graphs below demonstrate and give a general idea of league efficiency of pitchers versus their longevity in baseball games. Essentially, the more efficient the pitcher was found to be statistically, the further they tended to make it in the game they were pitching in. These graphs are for the sets over all 1,271,828 points, so there are some inaccuracies within their measures, primarily seen in inning 9.2. This could easily be fixed, and is in the experimental graphs shown below the 4, but for generalized graphing principals, the trends is clear.

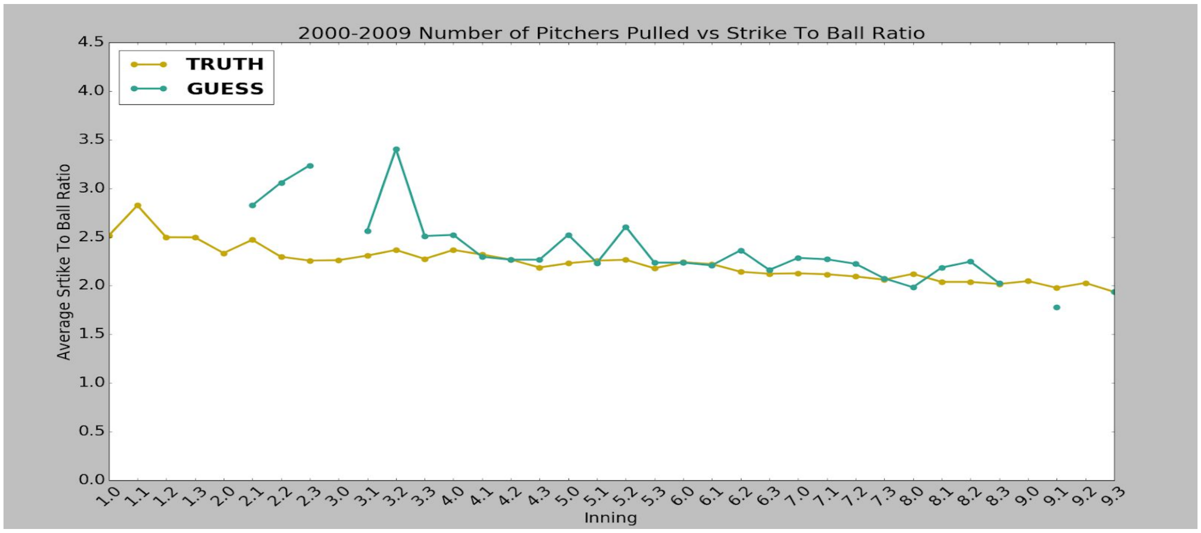

2000-2009 Pitchers Pulled By Inning vs Average Pitches Thrown vs Number of outs
Average Pitches Thrown vs Number of outs
4
5
6
7
8
9
10
11
Inning
1.0 1.1 1.2 1.3 2.0 2.1 2.2 2.3 3.0 3.1 3.2 3.3 4.0 4.1 4.2 4.3 5.0 5.1 5.2 5.3 6.0 6.1 6.2 6.3 7.0 7.1 7.2 7.3 8.0 8.1 8.2 8.3 9.0 9.1 9.2 9.3

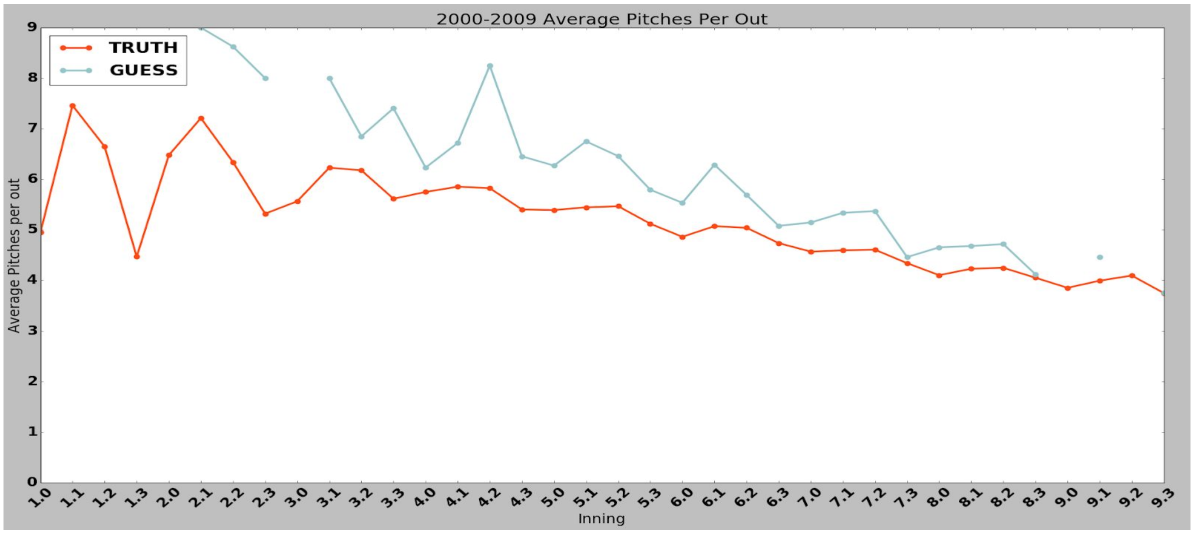

2000-2009 Pitchers Pulled By Inning vs Average Ball to Strike Ratio
Average Strike to Ball Ratio
1.8
2.0
2.2
2.4
2.6
2.8
3.0
Inning
1.0 1.1 1.2 1.3 2.0 2.1 2.2 2.3 3.0 3.1 3.2 3.3 4.0 4.1 4.2 4.3 5.0 5.1 5.2 5.3 6.0 6.1 6.2 6.3 7.0 7.1 7.2 7.3 8.0 8.1 8.2 8.3 9.0 9.1 9.2 9.3

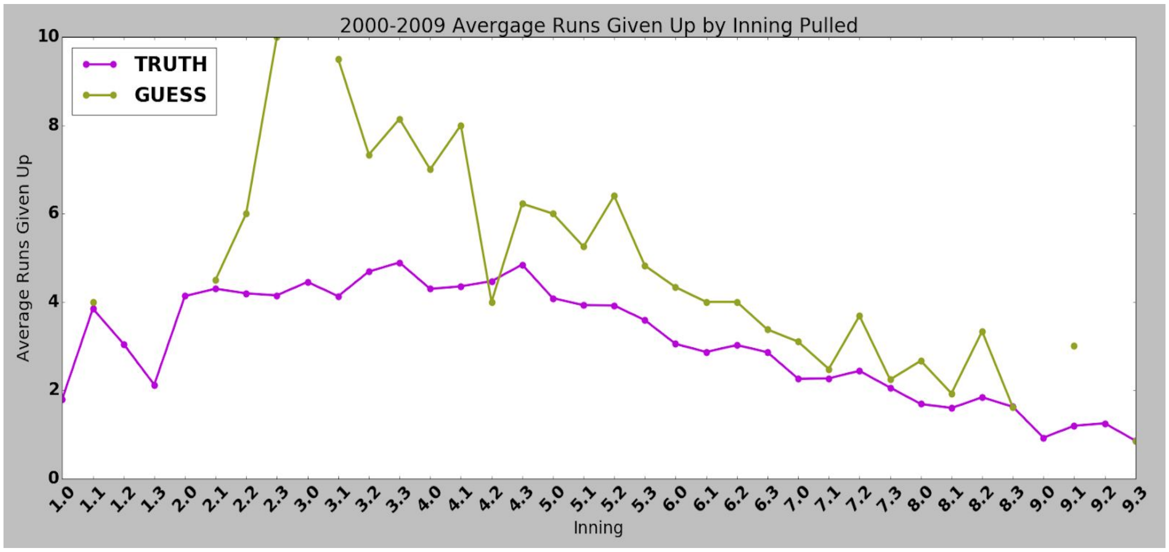

2000-2009 Number of Pitchers Pulled vs Inning and Outs
Pulled
0
1000
2000
3000
4000
5000
6000
7000
8000
Inning
1.0 1.1 1.2 1.3 2.0 2.1 2.2 2.3 3.0 3.1 3.2 3.3 4.0 4.1 4.2 4.3 5.0 5.1 5.2 5.3 6.0 6.1 6.2 6.3 7.0 7.1 7.2 7.3 8.0 8.1 8.2 8.3 9.0 9.1 9.2 9.3

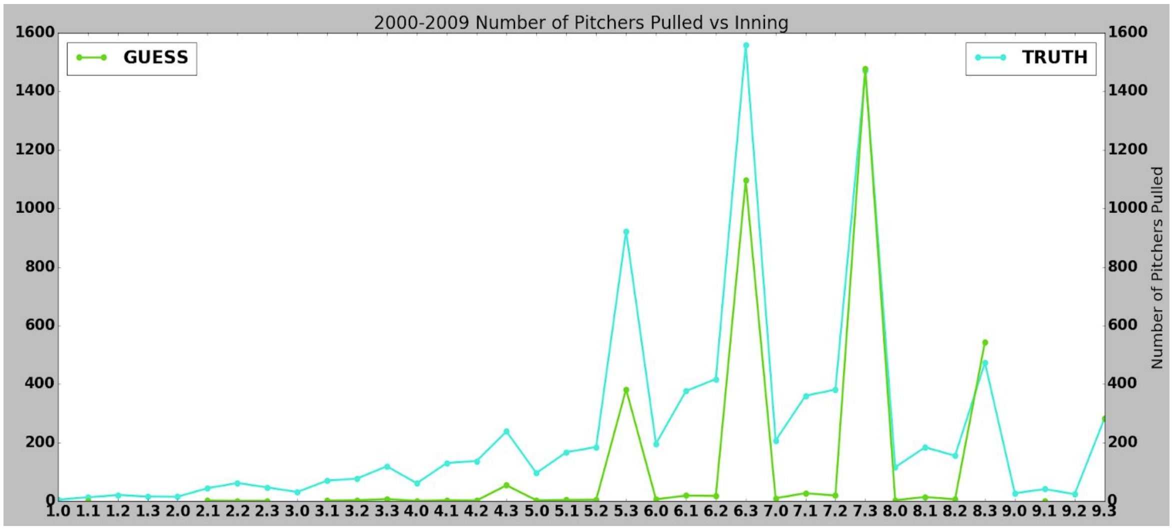


VIII. Experimental Graphical Representation









Looking at these graphs we can see that our prediction method is following the trend line very well. We can see that if there is something going a-typically wrong in baseball game or if there is an immediate call by our algorithm in a random-forest set to pull the pitcher. We can also see that the algorithm still does need to be refined however because for a typical pull, it appears that we are "low balling it.", especially in the 5th inning with 3 outs. However, for a datapoint where we have a decent number of statistical averages, say the 7th inning with three outs, we are dead on. Our model is also dead on for the 9th inning with 3 outs, because that is a given scenario where a pull is inevitable in a typical game that does not proceed to go into overtime.

VIII. Numerical Representation of Data

Here we show three different algorithms which we used to train our model. The experimental data above was generated using a random forest. See below for the graphical output of weighted values where the forest determined to be valuable information at which rate.

Random Forest with 100 estiators:

TPV: 0.713320
FPV: 0.973782
Total: 228075
TP = 2849
FP = 1145
TN = 218206
FN = 5875

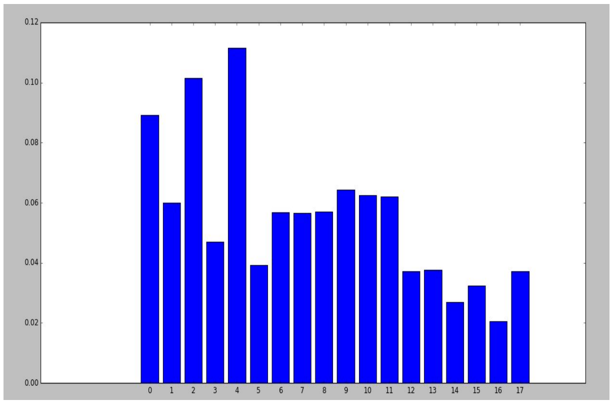
Total number of 1's guessed: 3994
Total number of 0's guessed: 224081
TP/TP+FN = 0.326570

K-nearest Neighbors with 501 neighbors:
TPV: 0.704733
FPV: 0.965662
Total: 228075
TP = 938
FP = 393
TN = 218958
FN = 7786
Total number of 1's guessed: 1331
Total number of 0's guessed: 226744
TP/TP+FN = 0.107519

Decision Tree with max depth 19:
TPV: 0.443788
FPV: 0.974793
Total: 228075
TP = 3154
FP = 3953
TN = 215398
FN = 5570
Total number of 1's guessed: 7107
Total number of 0's guessed: 220968
TP/TP+FN = 0.361531

Acronym	Meaning
TPV	True Positive/ True Positive + True Negative
FPV	True Negative / True Negative + False Negative
TP	True Positive
FP	False Positive
TN	True Negative
FN	False negative

Table 8.1



Each selection is ordered in the same order as table 1.

XI. Data We Believe Needs To Be Added

Data for Continuity:

Average Runs Given Up by Pitcher: (We don't want to use earned run average, because it shouldn't matter whether or not the run is "earned", all that matters is that it was given up.)
Average Innings Pitched by Pitcher
Average Innings Pitched by Team
Game Number for Team
Number of Games Pitched on the Season
Win-Loss Record

Data for Scenario:

Runs Scored by opposing team

Possible for Implementation, but needs to be considered

Errors by fielders
Earned Runs versus Non Earned Runs
Home or Away
Some sort of Bull-Pen Efficiency Metric
Strike-Contact Percentage (This would require RetroSheet and a complete Rework of Data Aggregation Methods. Worth Consideration)

X. Conclusion And Future Research

The original problem that we wanted to address was when the optimal time to pull a starting pitcher was. Though, as our research went on the problem shifted to predicting when managers would pull starting pitchers, and what factors determined this. With this in mind we were able to create a model that when it guessed a pitcher would be pulled, it was correct between 71%-74% of the time. This work could potentially be a tool to help managers in real-time. When using our model managers would be able to see on average what other managers would do from a 10 year aggregated data-set.

It would be worth considering a possible implementation where a Random Forest classifier is used on the entire league over the dataset, then defining the trendline of different teams, and how much their managerial style differs from the league average. Another possible implementation for the system would be a pitcher focused implementation where a league based average is used against a pitcher focused model and see how pitcher efficiencies and game outcomes are affected by straying from the recommendations of our model and league averages, essentially taking inspiration from Clifford Asness and Aaron Brown.[6]

[6] Pulling the Goalie: Hockey and Investment Implications, Asness C, Brown A, AQR Capital Management, 2018

References

1. Erickson B, et al, Exceeding Pitch Count Recommendations in Little League Baseball Increases the Chance of Requiring Tommy John Surgery as a Professional Baseball Pitcher, The Orthopaedic Journal of Sports Medicine, 5(3), 2017
2. Ganeshapillai, Gartheeban, and John Guttag. "A Data-driven Method for In-game Decision Making in MLB." *Sport Analytics Conference*. 2014
3. Donges, Niklas. "The Random Forest Algorithm – Towards Data Science." Towards Data Science. February 22, 2018. Accessed December 09, 2018. https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
4. Official Baseball Rules, MLB, p158 & 155, 2018
5. Gaines, C, Business Insider, MLB Ballpark Sizes Show The Immense Difference Between Fenway Park And Coors Field, 2014
6. Pulling the Goalie: Hockey and Investment Implications, Asness C, Brown A, AQR Capital Management, 2018