

Executive Summary

Baseball contracts involve intense negotiations and millions of dollars are at stake. Recently, the St. Louis Cardinals signed Paul Goldschmidt, agreeing to a five-year, \$130 million contract. Paul Goldschmidt is 32 years old. Age plays a big role in a baseball players performance. At some point as players get older, their performance on the field inevitably starts to decline. I have wondered whether the Paul Goldschmidt deal was good for the St. Louis Cardinals, and some have said, according to Forbes Magazine, there is reason to believe he could be entering the decline phase of his career. The Cardinals are betting that he will produce through the age of 37 years old. Will Paul Goldschmidt continue to perform through the age of 37? In more general terms, I would like to do analysis on MLB hitters and look at “m” years of past performance and predict the next “n” number of years of a major league baseball batter. Machine learning algorithms and predictive models will be used to facilitate the predictions. My customer in this analysis is baseball teams, analysts and fans.

What do future contracts look like for the next superstar position player of the game? Why a new model? Because the existing model is great for superstar players, but not so much for the owners. Fox Sports provided an article of 21 top contracts in baseball ([Top 21](#)). The average contract value was about 170 million dollars with an average contract duration of about nine years. Figure 1 shows the 21 players and their contract value.

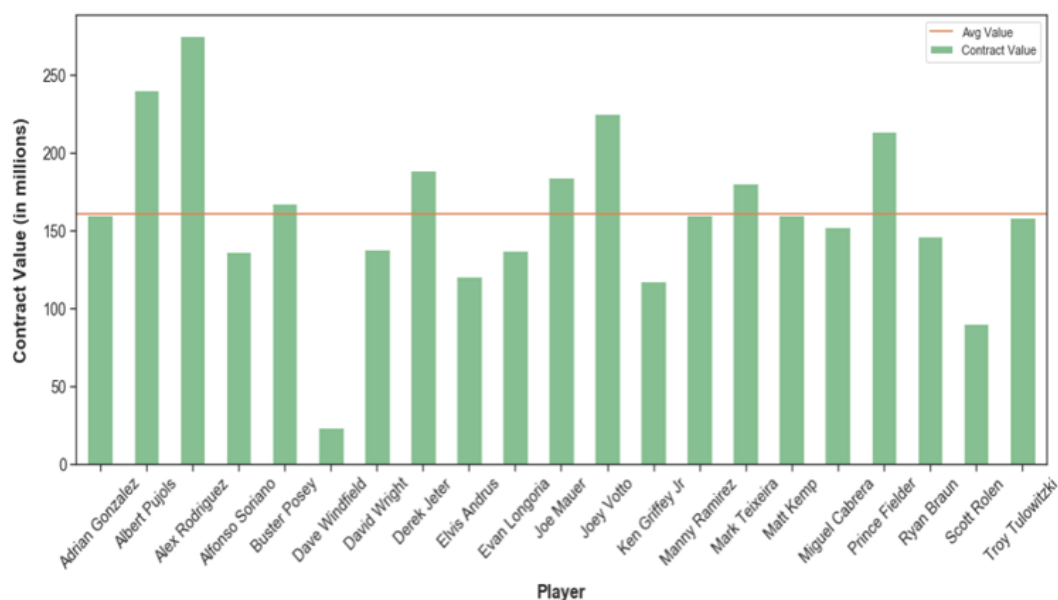


Figure 1 baseball position players – top contracts

Using this information and baseball statistics from Sean Lahman’s database, the 21 player’s combined career OPS performance was charted. The below chart plots the age of the 21 players vs. combined OPS performance. The average age at signing of the lucrative contracts was approximately 28 years of age and ended at about the age of 36. Note the trend after the contracts were signed. The owners who signed these players will likely not get a reasonable return on investment.

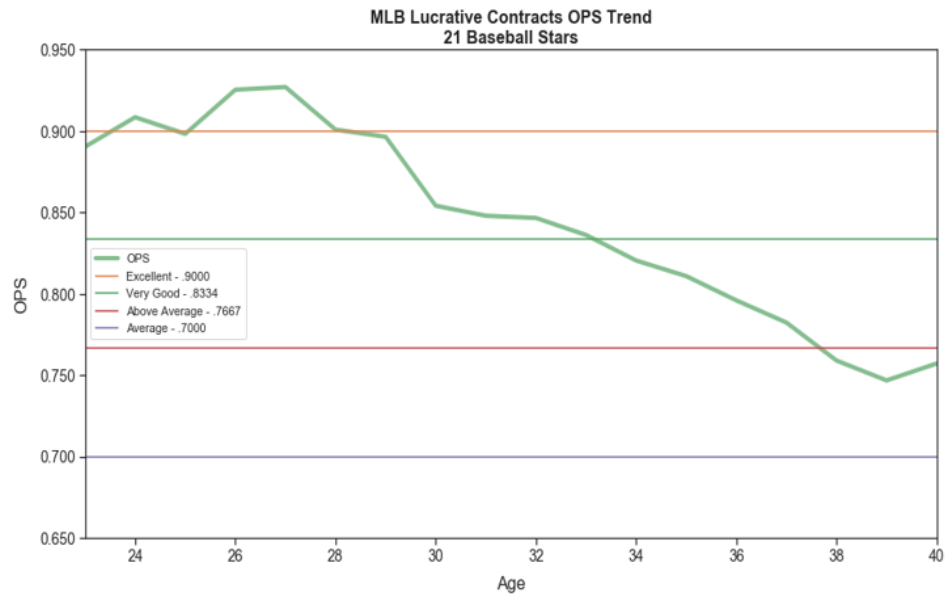


Figure 2 combined average OPS of 21 superstars OPS over time (age)

As shown in Figure 2, the contracts were signed at the peak of the players performance curve at around 28 years of age. As the player aged their performance gradually decreased at a rate of about 15 OPS points per year. Owners need a better model for signing players as figure 3 proposes.

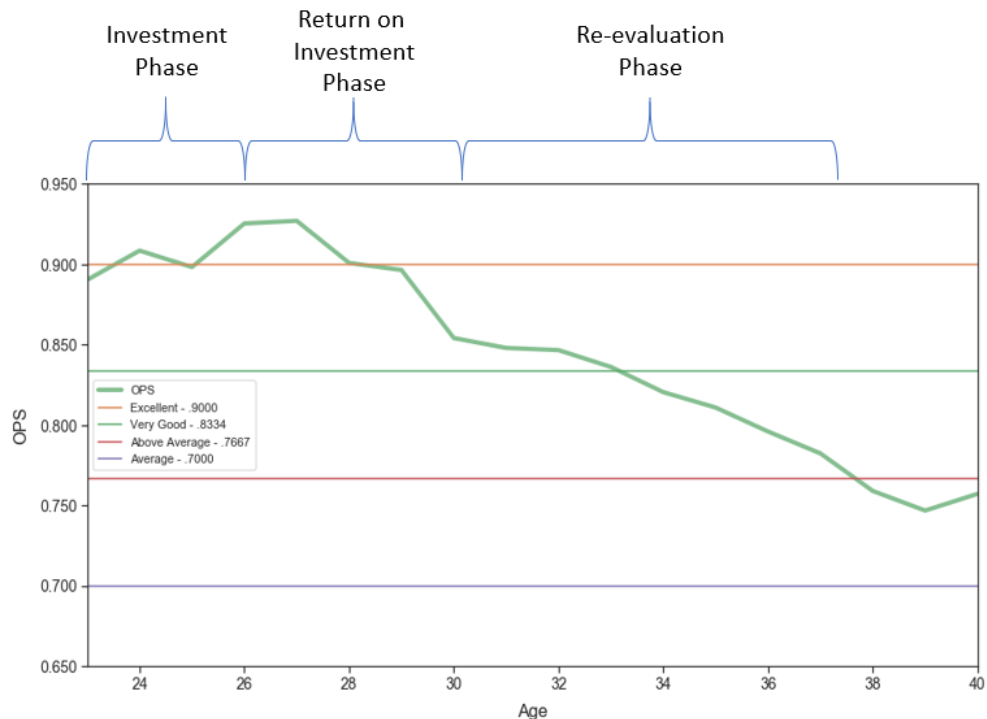


Figure 3 shows three phases of a position player's career from batting performance perspective.

In this model there are three stages: investment stage, return on investment stage and re-evaluation stage. The concept is to initially sign potential stars to a “fair” initial contract for 3 to 5 years. Advanced analytics and machine learning models should be able to reasonably predict whether the player will become that star given 3 to 5 years of performance data which will be shown as part of this project. Next, sign the player to a 5 to 6 years lucrative contract from age 26 to 31, and then re-evaluate the player after their lucrative contract is over. In this manner, owners get a return on investment on the millions invested in this player. That’s a tall order given player and agents may not agree with the approach. But it should be the ultimate goal. The goal of this project is to show that predicting a batter’s performance with reasonable certainty is possible given 3 to 5 years of major league performance statistics. Performance prediction in sports is challenging. There are so many variables in a player’s performance such as early aging, injuries, family problems, drug dependency problems, etc. Think about it, a 95 mile per hour fastball from 60 feet 6 inches (pitcher’s mound to home plate distance) arrives across the plate in .434 seconds. In that time, the player needs to decide whether to swing or not and make solid contact with a baseball that is about 3 inches in diameter with a bat that is about 3 inches in diameter. Crazy!

The remainder of this paper goes into the details on how the data was acquired, transformed and validated making it usable data, the process of exploratory data analysis (EDA) and finally statistical analysis and machine learning modelling and results.

Introduction

Major League Baseball has been America's sports pastime for over 100 years and was first founded in 1871 for the National League and 1901 for the American League. Today, there are 15 teams in each league. In the year 2000, the two leagues merged into what is now known as Major League Baseball (MLB). Since the very beginning, statistics in baseball has played a major role in the game. In today's baseball, advanced metrics are being used by every major league team in order to gain advantage over their competition. MLB organizations employ data science teams to collect this information for executives, general managers and coaches. But, statistics in baseball have always been polarizing. Some managers have lost their jobs recently because they could not adopt and did not believe in advanced metrics.

Quote from Bobby Bragan (baseball manager – 1940's) – "Say you were standing with one foot in the oven and one foot in the ice bucket. According to the percentage people, you should be perfectly comfortable."

Quote from Leonard Koppett (A Thinking Man's Guide to Baseball – 1967) – "Statistics are the lifeblood of baseball. In no other sport are so many available and studied so assiduously by participants and fans. Much of the game's appeal, as a conversation piece, lies in the opportunity the fans get to backup up opinions and arguments with convincing figures, and it is entirely possible that more American boys have mastered long division by dealing with batting averages than in any other way.

As ESPN Analyst Harold Reynolds said, "All of the sudden, it's not just BA and Runs Scored, it's OBA. And what is O-P-S?" Certainly, the "old standard" hitting metrics like batting average and runs scored have given way to more advanced metrics such as OPS (on-base plus slugging) which is a more meaningful metric on how well a player is performing at the plate. There are many other advance metrics today in baseball as well. I will be using OPS for this project. OPS is calculated by adding a player's on-base percentage with their slugging percentage.

The details of the equations are as follows:

$$\text{OPS} = \text{OBP} + \text{SLG}$$

$$\text{OBP} = (H + BB + HBP) / (AB + BB + SF + HBP)$$

$$\text{TB} = (n\text{Singles} * 1) + (n\text{Doubles} * 2) + (n\text{Triples} * 3) + (n\text{HomeRuns} * 4)$$

$$\text{SLG} = \text{TB} / \text{AB}$$

Where:

H – total number of hits of a player

BB – total number of walks (base on balls) of a player

HBP – total number of times the player was hit by a pitch

AB – total number of plate appearances (times at bat) by the player

SF – total number of sacrifice flies of a player

TB – is the total bases and is a weighted sum (1 for single, 2 for double, 3 for triple, 4 for HR).

So, TB is $(n\text{Singles} * 1) + (n\text{Doubles} * 2) + (n\text{Triples} * 3) + (n\text{HomeRuns} * 4)$ where nSingles is the number of singles, nDoubles is the number of doubles, nTriples is the number of triples and nHomeRuns is the number of home runs.

OPS – on-base plus slugging

OBP – on-base percentage

SLG – slugging percentage

NOTE: all statistics are taken over a period of time (typically a year)

(source : Wikipedia)

I would like to use OPS for both parts (performance/age prediction and PED usage) of analysis. In order to do this analysis, I have done some internet research and found raw baseball data collected from 1871 to 2018 of major league baseball games. All the above atomic data elements such as hits, at bats, etc are available and therefore OPS, OBP and SLG can be computed. Thanks to Sean Lahman and others, they have created a database with yearly baseball statistics from 1871 to 2018. The database has copyright 1996-2018 by Sean Lahman. I have read the license agreement which is licensed under Creative Commons Attribution and will not restrict me from using this data. The raw data needed will be from the 1954 to 2018.

Tony La Russa (ex St. Louis Cardinal Manager) has been quoted as saying (paraphrased) “you may not agree with me, but you don’t have all of the information that I have”. Now we do.

Predictive Model Details of Analysis

The discovery of Stein's Paradox in 1955 by Charles Stein of Stanford University undermined a century and a half of work on estimation theory (Morris, 1977). Stein's paradox concerns the use of averages to estimate unobservable quantities. From a baseball perspective, if you have 10 players and want to predict future batting averages of the 10 players, it is better to look at the 10 players as a whole, then to try to predict each person's batting average individually. And the future averages can be predicted no matter what the batting abilities of the players actually are. The process in Stein's method is to first calculate the average of the averages (grand average). Then shrink the individual averages towards the grand average. In other words, regression towards the mean (RTM). So, if one of the 10 players has a batting average above the overall league average then this player's average must be reduced. Alternatively, if one of the player's average is below the overall league average, then it must be increased. This shrunk value "z" is the James-Stein estimator. See the following link for additional information ([Stein's Paradox](#)). This project applied regression towards the mean to baseball batting performance statistic OPS (On Base Plus Slugging) and OBP (On Base Percentage). In 1970, Bradley Efron and Carl Morris applied Stein's methods to 18 major league baseball players. The figure below shows that the players true averages are clustered more closely around the grand average as predicted than was shown earlier in the season. The James Stein Estimator was used to predict the future state baseball batting averages of the 18 players. Even though the players initial averages were widely spread, in the end they clustered around the grand average as Stein's Paradox suggests.

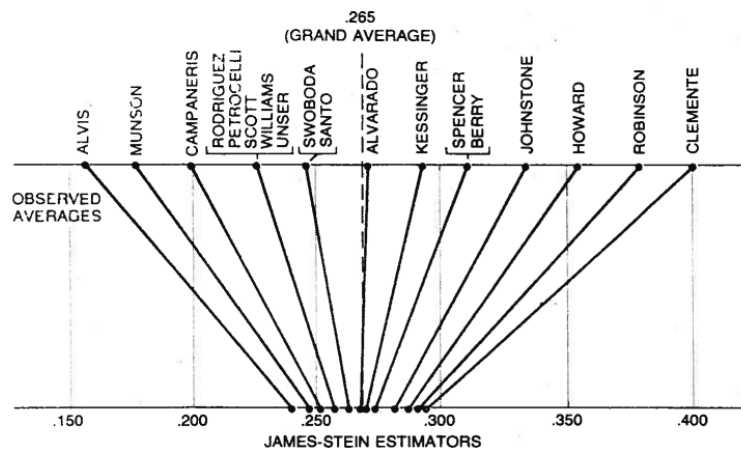


Figure was extracted from article by Efron and Morris of Stanford University

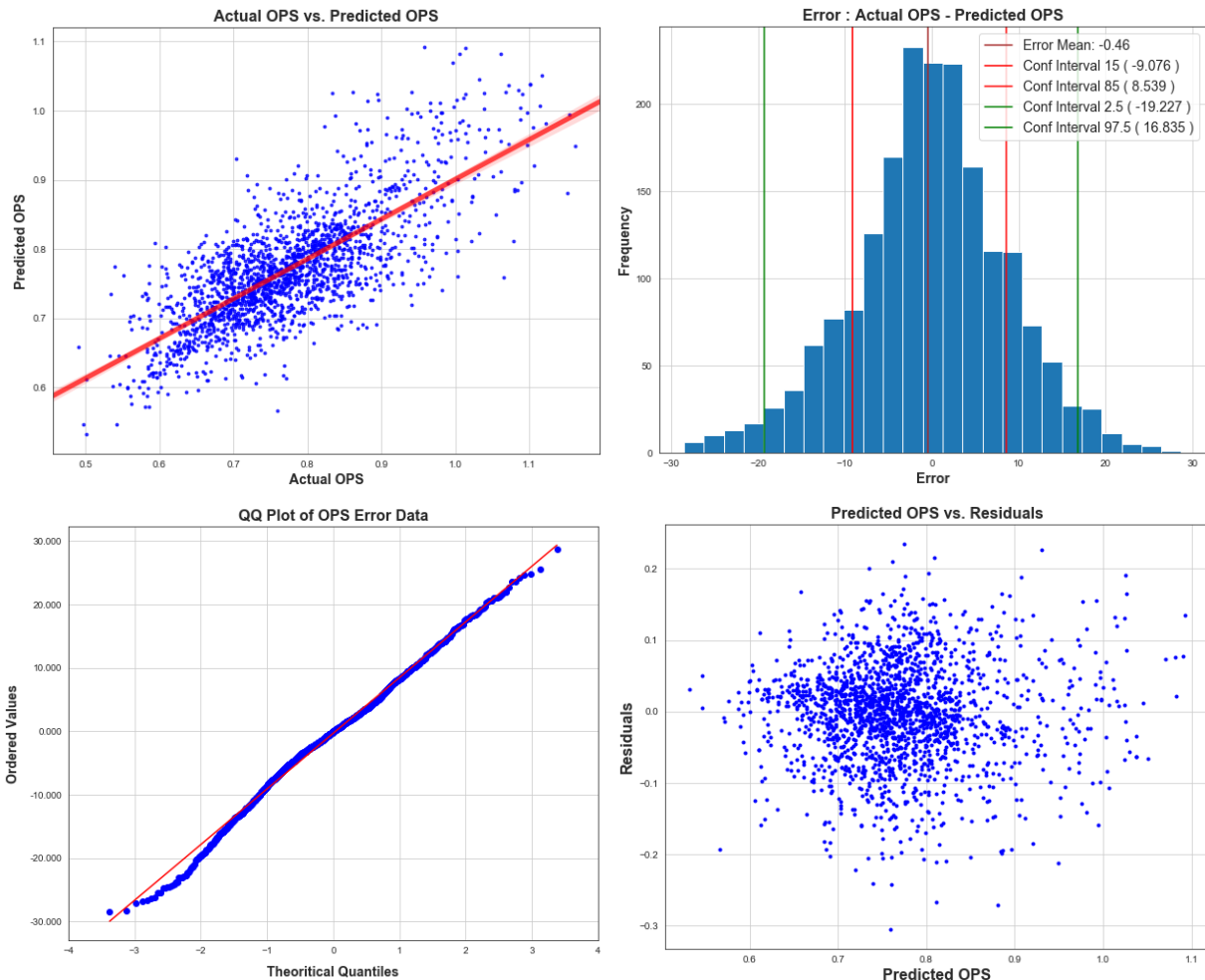
Data for this project comes from Sean Lahman's baseball data sets with batting performance collected from 1971 to 2018. Models were built using a set of features without the use of RTM and with the use of RTM. Various regression machine learning algorithms were implemented. The algorithms were Linear Regression, Ridge Regression, Lasso Regression, Non-Linear Regression, Random Forests, Support Vector Machines and XGBoost algorithms. See the results of all the runs by clicking on link below.

[GitHub Machine Learning Jupyter Notebook](#)

I have included the best run information below. XGBoost algorithm which stands for Extreme Gradient Boosting is a boosted tree algorithm and had the best score.

Here are the results from the run:

R Squared: 0.5668
Adjusted R Squared: 0.5632
F Statistic: 154.4975
MSE: 0.0049
RMSE: 0.0700
Test Observations: 1906
Sum of Abs Pct Error: 13124.9
Pct Mean Error: -0.4633
Pct Std Dev Error: 9.0641



There were three techniques that were considered for implementing the regression towards the mean which are 1) James Stein Estimator 2) Pearson's Correlation Coefficient 3) Bernoulli Trials approach. The Bernoulli Trials approach appealed to me because of its simplicity and flexibility. It seemed to fit well within the feature frameworks of my machine learning models and algorithms. Information on each of the estimators can be found as follows: James Stein Estimator ([Stein's Paradox](#)), Binomial Estimator [3-D Baseball](#) and Pearson Correlation Coefficient Estimator ([Pomona College](#)).

Regression towards the mean was applied to On Base Percentage (OBP) and to Slugging (SLG) individually. Then the two were added together to get to the regression to the mean OPS. The approach was to calculate cumulative mean values OBP and SLG percentages. For example, the

cumulative mean value of SLG was .419 (1954 to 1960) with a variance of .004643. **NOTE: only data from 1954 onward was collected due to lack of consistency of the treatment of Sacrifice Flies prior to 1954 even though data was available from 1871.** The cumulative mean value of OBP was .340 during this period with a variance of .001074. So, the cumulative mean value or grand mean of OPS was .759 in 1960. In 1970, cumulative SLG mean value was .409 (1954 to 1970), and the cumulative OBP was .334 resulting in a cumulative grand mean OPS of .743. These calculations were performed from 1960 through 2018 with data from 1954 to 2018.

To regress SLG to the mean, we need to calculate the number of at bats (AB) for which the binomial variance is the same as the variance of the true talent in the population sample.

$$\text{Observed Variance} = \text{True Talent Variance} + \text{Variance Due to Binomial Distribution}$$

or

$$\text{True Talent Variance} = \text{Observed Variance} - \text{Variance Due to Binomial Distribution}$$

The variance of the Binomial Distribution is so small due to the number of at bats (57,188) from 1954 to 1960. The variance due to binomial distribution with 57,888 at bats is .00000426 for SLG. True Talent Variance of SLG across all players from 1954 to 1960 would be .004643 - .00000426 = .004639. As the number of “at bats” increases, the variance due to binomial distributions gets very small and becomes negligible. For example, in 2018, the cumulative SLG variance is .005457 and the True Talent is .005457 because it is based upon a little over six million at bats from 1954 to 2018.

Now that we have the True Talent variance, we can find the number of at bats (n) which represents the True Talent Variance and plug it into the following equation noting that $\hat{p} \sim [p, p(1-p)/n]$

$$\text{TrueTalentVariance} = (p * (1-p)) / n \text{ or } n = (p * (1-p)) / \text{TrueTalentVariance}$$

$$n = (.419 * (1-.419)) / .004639 = \mathbf{52.5} \text{ at bats}$$

Next, the rtmTB needs to be calculated. To get this value multiply the SLG by the number of rtm at bats.

$$.419 * 52.5 = \mathbf{22}$$

For the rtmOBP, $\text{rtmPA} = n = (.340 * (1-.340)) / .001074 = \mathbf{209.8255}$ plate appearances; $\text{rtmOB} = 209.8255 * .340396 = \mathbf{71.4}$ on base events.

Next, these values need to be applied to all players in 1960. Let’s take an example from 1960 for Henry Aaron. The following table represents Henry Aaron’s statistics around SLG and OBP for the year 1960.

Henry Aaron

AB	H	1B	2B	3B	HR	BB	HBP	SF	OB	PA	TB	SLG	OBP
590	172	101	20	11	40	60	2	12	234	664	334	0.566	0.352

$$\text{SLG} = \text{TB} / \text{AB} = (1\text{B} + 2 * 2\text{B} + 3 * 3\text{B} + 4 * \text{HR}) / \text{AB}$$

$$\text{ActualSLG} = 334 / 590 = .566, \text{ rtmSLG} = (334 + 22) / (590 + 52.5) = .554$$

Performing the same calculation for OBP for 1960 and knowing that

$$\text{OBP} = \text{OB} / \text{PA} = (\text{H} + \text{BB} + \text{HBP}) / (\text{AB} + \text{BB} + \text{HBP} + \text{SF})$$

$$\text{ActualOBP} = 234 / 664 = .352, \text{ rtmOBP} = (234 + 71.4) / (664 + 209.8) = .350$$

Since Henry Aaron's stats are higher than the league average (AVG SLG = .419 and AVG OBP = .340), the regression towards the predicted SLG and OBP values are reduced.

We do this for all players for all years from 1960 to 2018 using the regression towards the mean values for each year.

Next, features (X) for our machine learning model were created along with OPS (y). We cannot use the current SLG and OBP actual and rtm values to predict the current year; so, we use the lag 1 values. So, for 2018, we use the 2017 SLG and OBP actual and rtm values. For 2017, we use the 2016 SLG and OBP actual and rtm value...and so on. After a substantial effort, the following are the features that were used for all model runs

- 'ndecade' – current decade for the year in which the player participated (zero mean normalized).
- 'nage' – age of the player for each year played normalized (zero mean normalized).
- 'nheight' – height of the player (zero mean normalized). It turns out weight doesn't affect the model.
- 'POS_1B' – bit map of 1 if player is a first baseman, 0 otherwise.
- 'POS_2B' – bit map of 1 if player is a second baseman, 0 otherwise.
- 'POS_3B' – bit map of 1 if player is a third baseman, 0 otherwise.
- 'POS_SS' – bit map of 1 if player is a short stop baseman, 0 otherwise.
- 'POS_OF' – bit map of 1 if player is an outfielder, 0 otherwise.
- 'lag1_rtm_nSLG' – previous year slugging with regression towards the mean (rtm) applied (zero mean normalized).
- 'lag1_rtm_ncSLG' – previous year slugging with rtm applied (zero mean normalized).
- 'lag1_rtm_nOBP' – previous year on base percentage with rtm applied (zero mean normalized).
- 'lag1_rtm_ncOBP' – previous year career on base percentage with rtm applied (zero mean normalized).
- 'lag1_rtm_nOPS' – previous year on base plus slugging with rtm applied (zero mean normalized).
- 'lag1_rtm_ncOPS' – previous year career on base plus slugging with rtm applied (zero mean normalized).
- 'lag1_rtm_nHR' – previous year Home Runs with rtm applied (zero mean normalized).
- 'lag1_rtm_ncHR' – previous year career Home Runs with rtm applied (zero mean normalized).

The machine learning algorithms run against the above model were: Linear Regression, Non-Linear Regression, XGBoost, Random Forests, Ridge Regression, Lasso Regression and SVM using regression towards the mean features. After each machine learning run, the results of the run by player by year are written to an excel spreadsheet for comparison purposes. Here are the results of the average actual OPS vs. Predicted OPS of the entire test set.

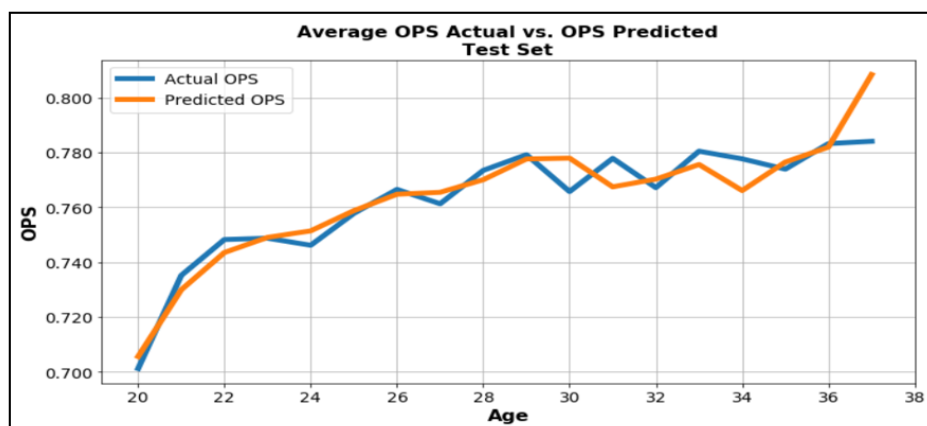


Figure 4 average actual OPS vs average predicted OPS by age

See the Jupyter Notebook for individual performance comparisons. In addition, a comparison was performed with non-regression toward the mean features (same features as above without rtm applied) and rtm features with the results of the comparisons below. For the comparisons, the two machine learning algorithms used were XGBoost and Ridge regression. The following are the R-Squared results of 10 runs using the XGBoost algorithm.

XGBoost: Regression Towards The Mean: R Squared Results (10 Runs)						
Run	Training Set		Testing Set			
	Lag1 Features	Lag1 RTM Features	Lag1 Features	Lag1 RTM Features	Diff	% Diff
1	0.579	0.559	0.507	0.525	0.018	3.4%
2	0.580	0.573	0.482	0.496	0.014	2.8%
3	0.593	0.576	0.458	0.483	0.025	5.2%
4	0.527	0.543	0.540	0.551	0.011	2.0%
5	0.593	0.579	0.477	0.493	0.016	3.1%
6	0.590	0.564	0.479	0.501	0.022	4.4%
7	0.548	0.571	0.482	0.510	0.028	5.5%
8	0.582	0.564	0.500	0.519	0.019	3.6%
9	0.540	0.561	0.529	0.551	0.023	4.1%
10	0.567	0.543	0.544	0.562	0.019	3.4%
Mean	0.570	0.563	0.500	0.519	0.019	3.8%
Std Dev	0.024	0.013	0.029	0.028	0.005	1.1%

XGBoost was used with GridSearchCV with CV=10
 20% held out for Test Set
 Training Set approx 8800 records (approx 4.5 million plate appearances)
 Testing Set approx 1600 records (approx 800,000 plate appearances)

Figure 5 XGBoost algorithm regression towards the mean feature comparison

The following are the results from using the Ridge Regression machine learning algorithm.

Ridge Regression: Regression Towards The Mean: R Squared Results (10 Runs)						
Run	Training Set		Testing Set			
	Lag1 Features	Lag1 RTM Features	Lag1 Features	Lag1 RTM Features	Diff	% Diff
1	0.456	0.497	0.466	0.508	0.041	8.1%
2	0.466	0.505	0.452	0.490	0.038	7.7%
3	0.473	0.511	0.424	0.468	0.045	9.5%
4	0.446	0.487	0.507	0.540	0.033	6.2%
5	0.467	0.507	0.448	0.482	0.034	7.0%
6	0.470	0.506	0.440	0.490	0.050	10.2%
7	0.468	0.504	0.447	0.498	0.051	10.3%
8	0.461	0.499	0.458	0.503	0.045	9.0%
9	0.458	0.495	0.488	0.534	0.046	8.7%
10	0.466	0.468	0.506	0.547	0.040	7.4%
Mean	0.463	0.498	0.464	0.506	0.042	8.4%
Std Dev	0.008	0.013	0.028	0.026	0.006	1.4%

Ridge Regression was used with GridSearchCV with CV=10
 20% held out for Test Set
 Training Set approx 8800 records (approx 4.5 million plate appearances)
 Testing Set approx 1600 records (approx 800,000 plate appearances)

Figure 6 Ridge Regression algorithm regression towards the mean feature comparison

As you can see from the testing set results, the regression towards the mean features did better than the features without regression towards the mean every time. XGBoost results were 3.8% better and Ridge Regression results were 8.4% better using rtm.

The models were run with a training set and a testing set with 80% training and 20% testing. A custom training / test split algorithm was used in favor of the standard `train_test_split()` function. Given a player's career, there should be no player that is split across the training / testing sets. That is, given a player, all yearly career statistics for this player should either be in the training set or the test set but not both. In this way, the training set does not see any years of a player who is being predicted in the testing set.

The training set input to the model consisted of all data from 1954 to 2018 excluding players who had less than 300 at bats (AB) in a given year. The purpose of this was to exclude pitchers who were included in the data as well as utility players and other players who did not have much playing time. My interest is players who are starters and play full time.

The testing set also excluded players with less than 300 at bats. In addition, analysis was performed and any player with an OPS of .300 or less or an OPS of 1.2 or greater was excluded as they fell outside two standard deviations of OPS values. Players with age of 19 or under and players 38 and over were also excluded as they fell well outside two standard deviations for player ages.

"Steroid Era" Analysis

The "steroid era" in baseball was a dark time for MLB baseball. Even today some of the greatest players of all time are not in MLB's Hall of Fame because of suspected steroid use. Mark McGuire is one such example. There is no exact start and end date to the steroid era. ESPN has defined it as starting in the later 1980's and ending in the late 2000's. Others have defined it to be from 1993 to 2003. The latter definition will be used for this project. In 2003, MLB introduced performance enhancing drug testing. Analysis of hitting performance of the pre-steroid era (1982 to 1992), the steroid era and post-steroid era (2004 to 2014) will be performed. I would like to know if the steroid era conclusively showed a significant advantage to player's performance using OPS as a metric. Figure 6 shows a times series plot with two fitted curves, one overfitted with polynomial degrees 40 and one properly fitted with polynomial degree 3.

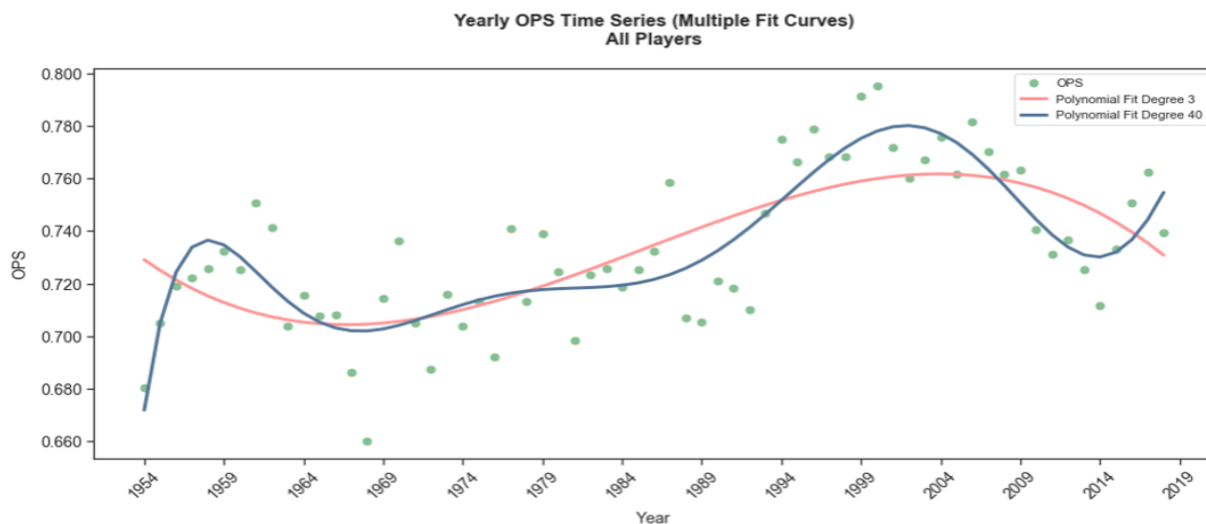
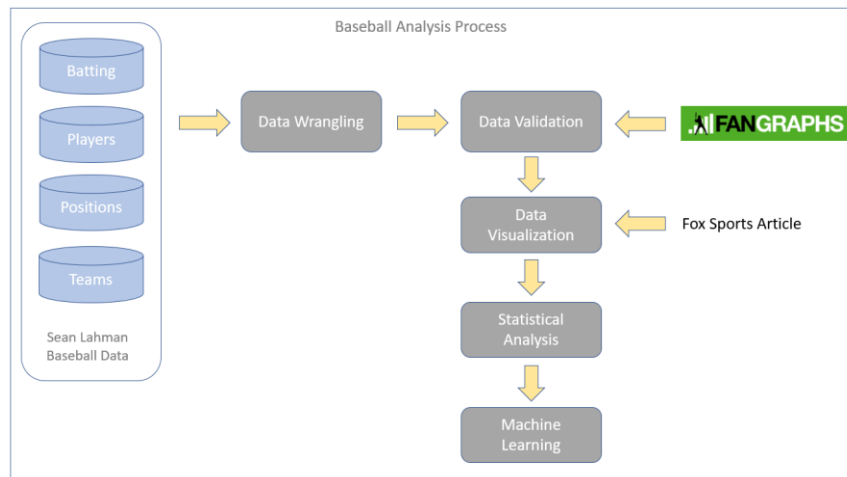


Figure 4 fitted curves to OPS trend data using overfitting and proper fitting curves

My customer is analysts who refuse to let players who took (or suspected of taking) performance enhancing drugs (PEDs) into the hall of fame because of unfair advantage. It is also for fans who want more information on the subject. Did they really have an advantage? A lot of work in this area has already been done, but I wanted to know for myself. Figure 3 shows an interesting trend. From about 1970, the OPS performance numbers gradually increased up until about 2009. They did not start in the steroid era, they started well before that. You could argue that starting in the early 1970's, the performance numbers gradually started to increase. In early 1990's it looks as if the performance numbers accelerated a bit. However, from my research, there isn't enough evidence to conclusively say the steroid era players had an advantage. The OPS performance numbers started increasing well before that era.

Stages of the End-To-End Process

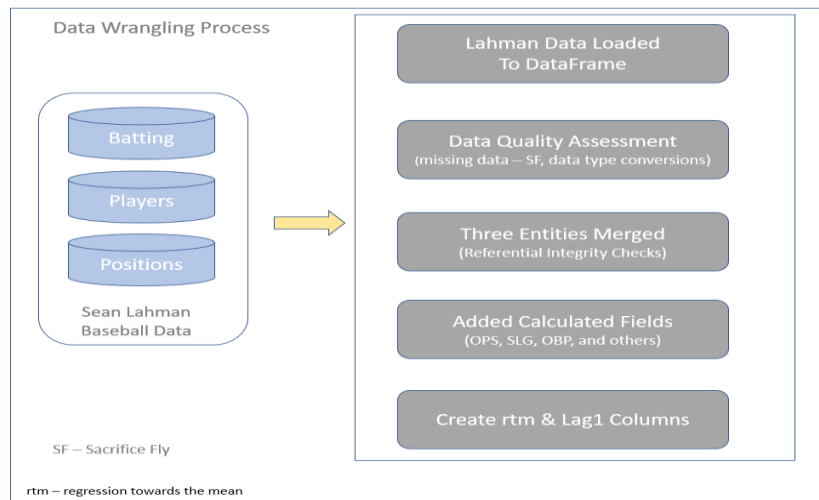
There are five main components of the baseball analysis process:



The five steps are Data Wrangling, Data Validation, Data Visualization, Statistical Analysis and Machine Learning.

Data Wrangling

The first step is Data Wrangling. Data was downloaded from the Sean Lahman site and staged for loading. There were three main data entities that were loaded: Batting, Player and Position data. Why only use data from 1954 when data was available from 1871 onwards? The sacrifice fly was not tracked consistently until 1954. According to Wikipedia, “batters have not been charged with a time at-bat for a sacrifice hit since 1893, but baseball has changed the sacrifice fly rule multiple times. The sacrifice fly as a statistical category was instituted in 1908, only to be discontinued in 1931. The rule was again adopted in 1939, only to be eliminated again in 1940, before being adopted for the last time in 1954”. Sacrifice flies are required for OPS calculation, and for this reason only data from 1954 onward is used. Here is the process for Data Wrangling:



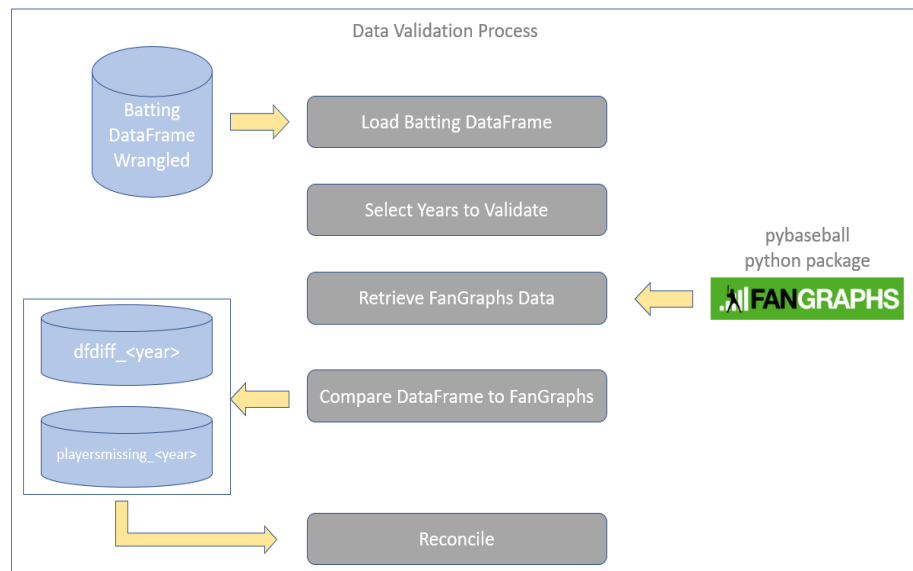
A data quality assessment was performed, the three entities were joined, additional columns were added and the DataFrame was written to a CSV file to be used in downstream processes. Here is a link to the code bases:

[GitHub Data Wrangling Code](#) , [Data Wrangling rtm Code](#) , [Data Wrangling Lag Code](#)

If you get an error, it is likely caused by “GitHub API rate limit exceeded”. You’ll have to try later.

Data Validation

The next step in the process is Data Validation. The following diagram defines the data validation process. In order to independently validate the Lahman data after all the data wrangling was performed, the FanGraphs API was used. The pybaseball package integrated the FanGraph API. All that was needed was a function call which implemented the API which was provided by the pybaseball package.



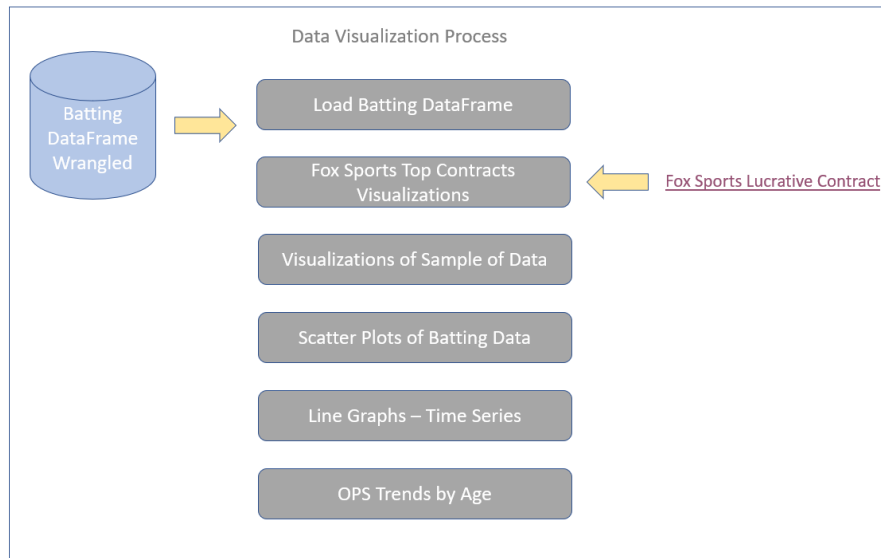
All the wrangled Lahman data was successfully reconciled using FanGraphs. Here is a link to the Data Validation code:

[GitHub Data Validation Code](#)

If you get an error, it is likely caused by “GitHub API rate limit exceeded”. You’ll have to try later.

Data Visualization

The third step in the process was Data Visualization. During this step, EDA was performed. To make it interesting, a Fox Sports article listed the top contracts in MLB which listed the dollar amount of the contract, the duration of the contract and when it was signed. This data was manually entered into a spreadsheet, loaded and integrated with the wrangled Lahman data. The following flows the overall data visualizations performed:



Here are the links to the Fox Sports article and the link to the data visualization Jupyter Notebook.

Fox Sports Article:

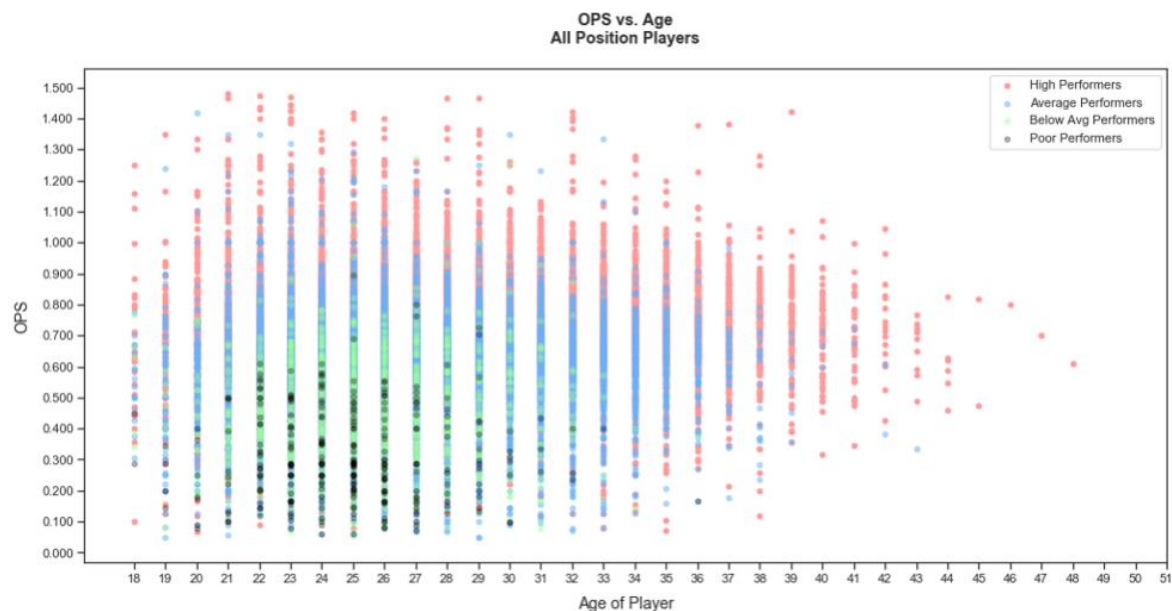
[Fox Sports MLB's Most Lucrative Contracts](#)

Jupyter Notebook – Data Visualizations

[GitHub Exploratory Data Analysis](#)

If you get an error, it is likely caused by “GitHub API rate limit exceeded”. You’ll have to try later.

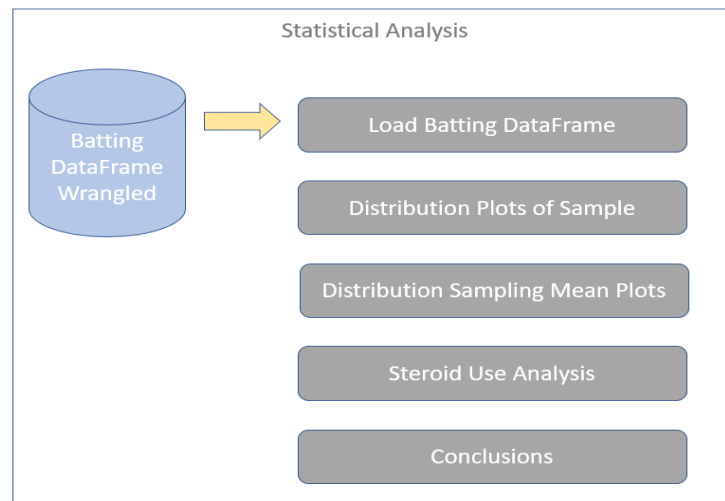
Below is the scatter plot of all MLB players from 1954 to 2018. Note the bands of colors representing different categories of players.



Initially, with one single color it was very difficult to visualize anything. You can already observe in the color bands that there is an upswing in performance of each category of player and then a downswing of performance over their career.

Statistical Analysis

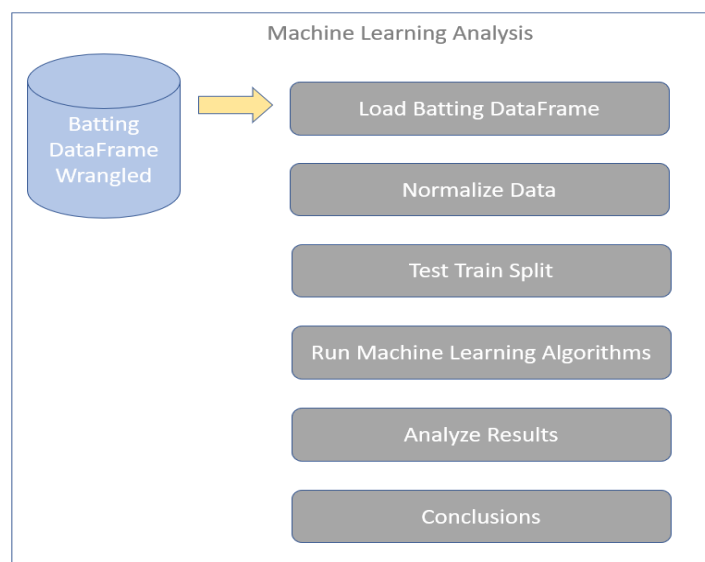
To further my understanding and answer my questions, statistical analysis was performed. The following summarizes the analysis steps as follow:



The Jupyter Notebook with the full analysis can be found by clicking on the following link: [GitHub Statistical Analysis](#)

Machine Learning Analysis

The machine learning process is as follows:



The initial process starts the loading of the wrangled batting data in csv format into a dataframe. After that features are normalized, and the test train sets are created from a custom test train split function. After that the machine learning algorithms are run and analysis is provided as to the success of each run. The following link takes you to the GitHub Jupyter Notebook for all of the machine learning runs.

[GitHub Machine Learning Jupyter Notebook](#)

There are three charts that I would like to highlight as part of the statistical analysis: Yadier Molina, Albert Pujols and Paul Goldschmidt performance plots.

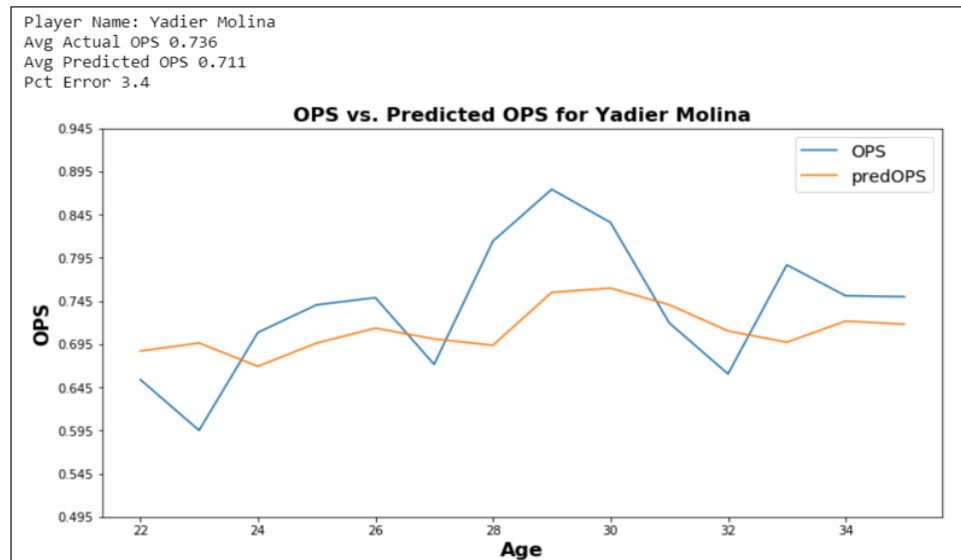


Figure 7 actual OPS vs. predicted OPS for Yadier Molina

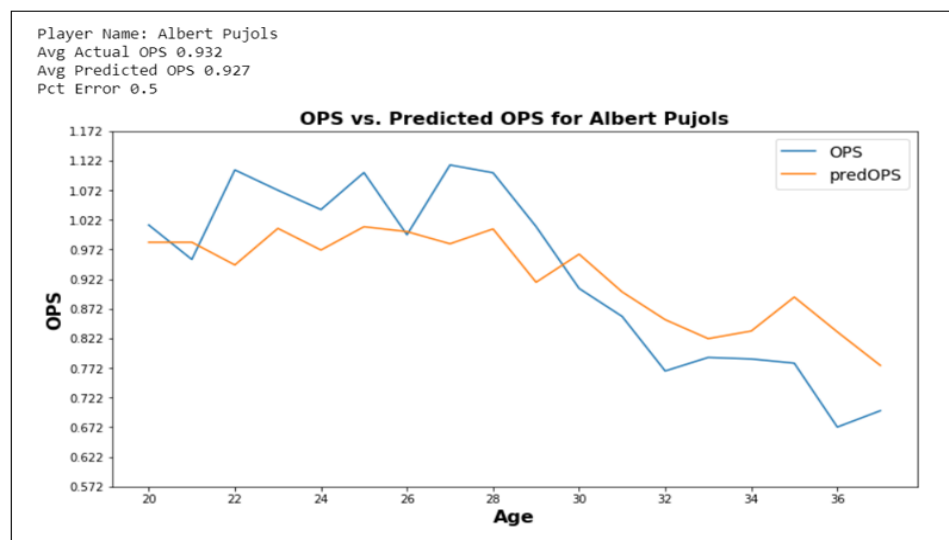


Figure 8 actual OPS vs. predicted OPS for Albert Pujols

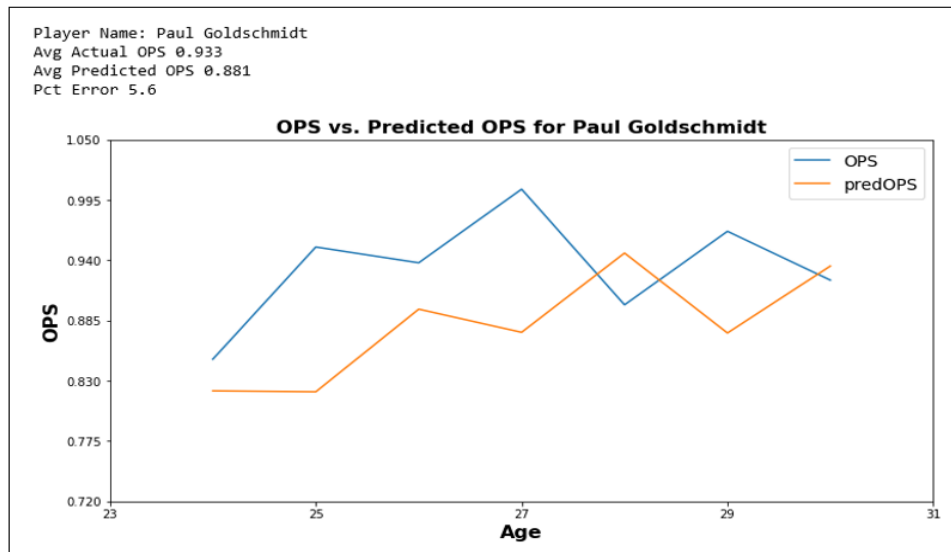


Figure 9 actual OPS vs. predicted OPS for Paul Goldschmidt

The project showed that you can predict future batting performance using OPS. But there are limitations to how accurate you can be. There are too many variables that we have no control over which leads to somewhat varied performance from year to year. The regression towards the mean proved to be an important part of predictions as it added around 4% to 8% increase in R-Squared values in XGBoost and Ridge Regression. The predictions tend to under predict batting performance which is a good thing. If it had tendencies to over predict, then a decision might be made that a player will have a .900 OPS when it will actually be .800. I would have it underpredict as a conservative approach.