Team 89: Final Report

Introduction - Motivation

Today, data science is being used heavily to predict player performances in the MLB. This paper develops an analytical approach to discover baseball players most likely to get a hit on any given day in order to get the longest possible hitting streak.

Problem Definition

The objective is to develop an application that utilizes sabermetrics and machine learning to discover players that are most likely to get a hit on any given day. and ensuring that our model has a higher accuracy than current methods like Random Guessing (~60%) or Selections from Top 4 Picks (~67%).

Literature Survey:

Our project touches multiple areas of interest such as comparison of various matchups, a variety of algorithms such as SVM and Random Forest, and most importantly hit prediction of batters. Here is some of the existing research we reviewed for this project:

Using multi-class classification methods to predict baseball pitch types[1]

Main Idea

Predictions of the next type of pitch based on prior pitches using machine learning models, random forest was the most successful.

Useful for Project

Introduction of the Pitchf/x database, and suggestions of machine learning algorithms that work well.

Potential shortcomings

The scope might be too narrow, broadening and using a more general dataset might be helpful.

Matchup models for the probability of a ground ball and a ground ball hit[2]

Main idea

Examines likelihood of a batter to hit a ground ball for a given matchup.

Useful for Project

Introduces the HITf/x database, and introduces the concept of general characteristics of pitchers and batters.

Potential shortcomings

Emphasizes single variable importance rather than multivariate statistical or machine learning models.

Predicting Win-Loss outcomes in MLB regular season games – A comparative study using data mining methods[3]

Main idea

Uses four different machine learning algorithms in order to build predictive models of regular season wins for MLB teams.

Useful for Project

Introduces and discusses many issues around data quality.

Potential shortcomings

A lot of focus on the methodology rather than its predictive ability.

Predicting MLB Player Performance Using Decision Trees[4]

Main idea

Use decision trees to predict MLB player performance.

Useful for Project

Useful if we use a decision tree model. The features that they used are also good to know.

Potential shortcomings

The authors state that they were vulnerable to overfitting at times. We will do everything we can to reduce overfitting in our model(s).

Maximizing Precision of Hit Predictions in Baseball[5]

Main idea

The authors analyzed different models that produced the player most likely to get a hit on any given day. The tested models include logistic regression and an SVM.

Useful for Project

The results from each model are useful benchmarks for our model.

Potential shortcomings

Their SVM was overfitted. They also revealed additional feature ideas that should be tested.

Comparison of Methods for Batter-Pitcher Matchups[6]

Main idea

Discusses various methods for predicting baseball matchup outcomes.

Useful for Project

Useful for deciding which method(s) to use.

Potential shortcomings

The author suggests using a combination of the three reviewed methods and weighing their outcomes appropriately.

Workforce Analytics in Baseball Player Management[7]

Main idea

Introduces Expected Performance Efficiency (EPE) using optimization on offensive and defensive value of players, health and injury data and player salary.

Useful for Project

Useful data sets and techniques to compute performance value of a player.

Potential shortcomings

Relies heavily on Wins Above Replacement (WAR), while ignoring the match between individual personality traits and clubhouse characteristics.

Bias in the log5 estimation of outcome of batter/pitcher matchups, and an alternative[8]

Main idea

Introduces limitations in log5 estimation for batter/pitcher matchup results and presents Morey-Z as an alternative.

Useful for Project

Useful to predict batter/pitcher metrics like Home Run Percentage with probabilities well below 0.5

Potential shortcomings

The effects of log5 bias are small and difficult to identify when the probability statistic is close to 0.5 and normally distributed.

Using Machine Learning Algorithms to Identify Undervalued Baseball Players[9]

Main idea

Using ML Clustering to find undervalued players and projecting player value in a subsequent season.

Useful for Project

Useful to build best value teams and/or helping front office find key players.

Potential shortcomings

Requires sufficiently large amount of data about each player to include in the analysis.

A Monte Carlo Approach to Joe DiMaggio and Streaks in Baseball[10]

Main idea

Through Monte Carlo simulation, it is proved that a 56 game-hitting streak is not unlikely to happen.

Useful for Project

DiMaggio's achievement was the basis of the "Beat the Streak" contest and a Monte Carlo approach could be useful for estimating the probabilities of getting a hit.

Potential shortcomings

Certain information would be interesting but hard to take account of.

Analyzing Baseball Statistics Using Data Mining[11]

Main idea

Develops a model that predicts the performance of teams in the playoffs.

Useful for Project

The data collection and preprocessing is complete and well explained; could be a basis for our own.

Potential shortcomings

It was done with a small amount of data, which led to a model with medium accuracy.

Machine Learning Applications in Baseball: A Systematic Literature Review[12]

Main Idea:

A review about different approaches of machine learning applications on baseball.

Useful for project:

The list of articles on this review could be useful for knowing which approaches have worked out the best.

Potential shortcomings:

The listed models that predict hits is small.

Beating the MLB Moneyline[13]

Main Idea:

Linear regression, SVM and random forest approaches to create betting strategies that can win the MLB moneyline.

Useful for project:

This helps us get an idea on feature selection.

Potential shortcomings:

Moneyline odds and older season statistics were not taken into consideration.

Predicting the Final Score of Major League Baseball Games[14]

Main Idea:

Monte Carlo simulations on a Markov Decision Process to predict outcome of baseball games.

Useful for project:

The Return on Investment approaches are good for maximizing the profitability

Potential shortcomings:

Compute intensive to a model and cross validate with lot of data.

Applying Machine Learning Techniques to Baseball Pitch Prediction[15]

Main Idea:

SVM and KNN models to predict pitch types.

Useful for project:

Idea of using different optimal set of features for each pitcher/count pair.

Potential shortcomings:

Outcome is just a binary classification, if the next pitch is fastball or not.

Streaky Hitting in Baseball[16]

Main Idea:

Method to measure how streaky a player is given player's ability and number of hitting opportunities.

Useful for project:

Definition of streakiness is useful to determine when a player is likely to get a hit.

Potential shortcomings:

Only accounts for batter, ignores the impact of the pitchers and ballpark in the model.

zWins, an Alternative Calculation of Wins Above Replacement in Baseball[17]

Main Idea:

A new metric to measure how much better than replacement a player is, replacing WAR.

Useful for project:

One metric, zOffense, is used to calculate runs above replacement for offensive production.

Potential shortcomings:

Not isolated to hits: stealing bases, getting walked, scoring runs, and other non-hits are accounted towards winning.

Beat the Streak[18]

Main Idea:

Build an ML model that predicts which player is most likely to get a hit on a given day.

Useful for project:

Show features that go into model to predict individual player performance each day.

Potential shortcomings:

Does not use advanced stats for model, only accounts for starting pitcher, relies heavily on short term stats.

Proposed method

Intuition

This algorithm should be better than the current state of the art because we have taken the peer-reviewed methods proposed by researchers and added on to them. The algorithm first starts by looking at the "Hot Batter Matchups". This is one of the current ways that most entrants select a batter for the Beat the Streak as it looks at batters that have performed well against their pitching matchups on a given day and establishes our floor. We then take the features from the "Hot Batter Matchups" and train both a Random Forest Classifier and a Logistic Regression classifier using data from Baseball reference to create our ensemble learner.

Rather than relying on a single model that could result in an outlier result on a given day, we increase our confidence with our other model. One of our biggest innovations in these algorithms is to calculate the probability per at-bat rather than per-game. Intuitively, a more granular probability for each batter gives us more confidence than a per-game probability since we are more precise in our calculations. We do this for both models and then set minimum thresholds and only pick a batter (or two!) if we have a certain level of confidence. If no player meets these criteria, then we do not have enough confidence to make a pick and punt for the day. Therefore, we believe that the algorithm we have developed is better than the state of the art of simply selecting the hottest batters or picking the batters with the best matchups.

Description

Initially, we intended to use a Random Forest Classifier as the foundation of our analysis. However, due to the way that our training data was structured, we had to switch to regression entirely. Our models currently predict the probability that a player has of getting a hit per at-bat. This just means that players can't be classified properly on a per game basis. Our current method of identifying Beat the Streak picks involves downloading a csv file from RotoWire that contains "Hot Batter Matchups". These list batters who have performed well against the starting pitcher for that given day and the matchup stats between the two. We trained our models on the very same features that we acquire from this csv for analysis.

Once our Random Forest and Logistic Regression assign a hit probability for each matchup, we compare the results to see if they can agree on a top player for the day. If they each report the same player in their outputs and his average hit probability > .285, he becomes our pick for Beat the Streak. If they each report two or more of the same players in their outputs with average hit probabilities > .285, this signals a “double down” day where we select the two with the highest average hit probabilities. If they can’t agree on any players, then it is a “no pick day”.

Data Preparation

The raw data for this experiment was obtained from Baseball Reference Database. A conscious effort was made to clean up and transform data, handle null values and select appropriate features useful for hit analysis. The model is trained on data from 2016 to 2018, with a total of over 480,000 records. The following features were computed for each batter/pitcher pair:

- At-Bat: number of times batter has faced pitcher
- Hits against pitcher
- Extra Base Hits against pitcher
- Home Runs against pitcher
- Strike Outs against pitcher
- Run Batted in: Runs scored as a result of batter's plate appearance
- Slugging Percentage: Total bases/At-bats
- Ballpark: Where the game will be played
- Batting Average: Hits/At-bats against the opposing pitcher

Data Analysis:

We built a Random Forest and Logistic Regression model using scikit-learn. Our models were trained using the prepared dataset described above. We tuned the hyperparameters using scikit’s built in GridSearchCV and a wide range of hyperparameter options. Our Random Forest has 400 estimators, a max depth of 5 nodes, and requires at least 20 samples to be a leaf node. Our Logistic Regression model is trained with the Limited-memory BFGS solver, fits a binary problem for each label since this is a binary problem, and has 200 max iterations. They are currently reporting ~75% testing accuracy. We have made efforts to optimize the accuracies and will continue to do so throughout phase three. Details regarding our experiments can be found below.

User Interface

We created a user interface using Pelican, a web framework written in Python. The user interface is hosted on Github pages, and incorporated Jupyter Notebooks for visualization. The interface took the form of a blog, in which the user can interact with tables and charts that display the results of our analysis each day. Blogging is a popular method of showing and visualizing the results of sports analytics studies. We began updating the blog posts every day and recommending picks for the Beat the Streak competition in mid-April. In addition, when our algorithm did not give conclusive reason to pick a player, we created interactive charts showing each algorithm’s performance through history. The site is hosted at https://seaminglyaccurate.io

Experiments/ Evaluation

Description

We ran a few experiments in order to optimize our model and determine our testing accuracy.

GridSearch/Cross-Validation:

Hyperparameter tuning was performed on both of our models to optimize training/testing accuracies. The following parameters were tuned for our models:

Random Forest:

- n_estimators
- max_depth
- criterion
- min_samples_leaf

Logistic Regression:

- solver
- max_iter
- multi_class

Feature Importance:

We analyzed the importance of each feature using the feature importance function:

Ballpark (0.287810)

Slugging Percentage (0.167103)

At-Bat (0.163942)

Batting Average (0.112588)

Strikeouts (0.092533)

Runs Batted In (0.080904)

Hits (0.039907)

Extra Base Hits (0.033425)

Homeruns (0.021788)

After adding samples from 2018, we ran the feature importance function again:

Ballpark	(0.298418)
Batting Average	(0.230572)
At-Bat	(0.159503)
Strikeouts	(0.141555)
Slugging Percentage	(0.085434)
Hits	(0.031991)
Runs Batted In	(0.030433)
Homeruns	(0.011642)
Extra Base Hits	(0.010452)

As you can see, the weight that the random forest put into slugging percentage took a big hit. Batting average against the opposing pitcher is now considered one of the most important factors in hit probability.

Principal Component Analysis:

We used PCA to reduce the number of features while maintaining as much of the variance as possible. We can explain over 98% of variance by keeping the first two principal components. We then ran the previous experiments on the reduced data and achieved similar accuracy. This means that we can transform our dataset into the top two principal components and maintain confidence in our predictions.

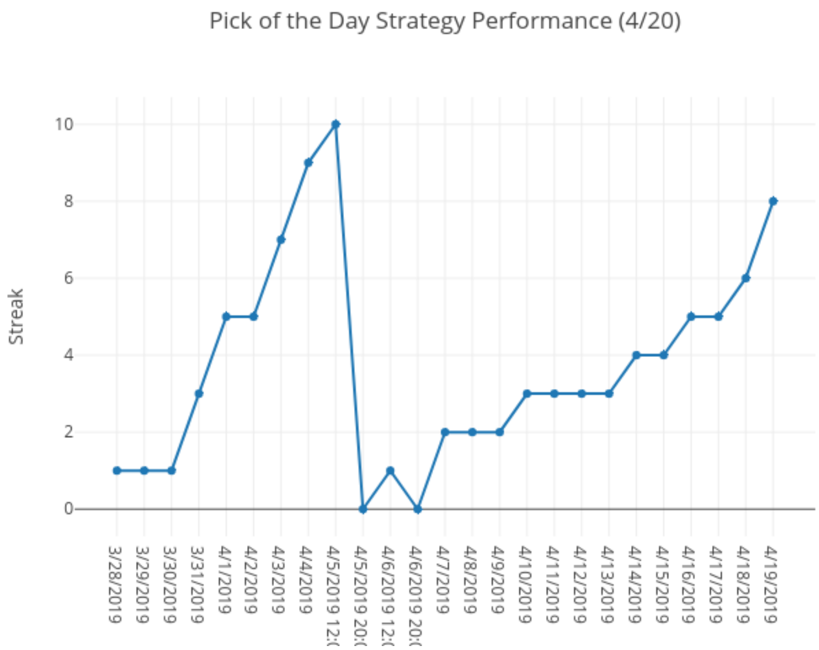
Conclusions and discussion

Our Random Forest went 15/22 (68.2%) with its highest streak getting to 6. Not very good, but since we added 2018 data and re-tuned it, it has gone 9/11 (81.8%).

Our Logistic Regression model went 18/22 (81.8%) with its highest streak getting to 8.

Our program went 19/21 (90.4%) with its highest streak getting to 10 and its current streak at 8. This is truly remarkable and proves that using our models together with a formulated pick strategy will yield better results than using either model separately.

Ultimately, our project was a tremendous success, because it is more accurate than Random Guessing (~60%) or Selections from Top 4 Picks (~67%). Below is a plot of the results.



Distribution of team member effort:

All team members have contributed similar amount of effort.

References:

[1]: Siddle, Glenn, and Hien Tran. “Using Multi-Class Classification Methods to Predict Baseball Pitch Types”. Journal of Sports Analytics, 27 Feb. 2018, https://content.iospress.com/articles/journal-of-sports-analytics/jsa171

[2]: Healey, Gleen. “Matchup Models for the Probability of a Ground Ball and a Ground Ball Hit”. Journal of Sports Analytics, 11 Apr. 2017, https://content.iospress.com/articles/journal-of-sports-analytics/jsa0025

[3]: Soto-Valero, César. “Predicting Win-Loss outcomes in MLB regular season games – A comparative study using data mining methods”. International Journal of Computer Science in Sport, Dec. 2016, https://www.researchgate.net/publication/311862823_Predicting_Win-Loss_outcomes_in_MLB_regular_season_games_-_A_comparative_study_using_data_mining_methods

[4]: Ruegg, Kurt, Paluri, Srujana and Chou, Amy. “Predicting MLB Player Performance Using Decision Trees”. Santa Clara University. http://www.cse.scu.edu/~mwang2/projects/Predict_MLBplayerPerformance_16w.pdf

[5]: Clavelli, Jason and Gottsegen, Joel. “Maximizing Precision of Hit Predictions in Baseball”. Stanford University. 13 Dec. 2013, http://cs229.stanford.edu/proj2013/writeup.pdf

[6]: Thakur, Siddhartha. “Comparison of prediction methods for batter-pitcher matchups”. The University of Texas at Austin. May. 2016, https://repositories.lib.utexas.edu/handle/2152/45747

[7]: Michael Greene, Adam Hirsch. “Workforce Analytics in Baseball Player Management”. MIT Sloan Sports Analytics Conference 2011. 4-5 Mar. 2011, http://www.sloansportsconference.com/wp-content/uploads/2011/08/Workforce-Analytics-in-Baseball-Player-Management.pdf

[8]: Moreya, Leslie C. and Cohen, Mark A. “Bias in the log5 estimation of outcome of batter/pitcher matchups, and an alternative”. Journal of Sports Analytics. 2015, https://content.iospress.com/download/journal-of-sports-analytics/jsa0005?id=journal-of-sports-analytics%2Fjsa0005

[9]: Ishii, Tatsuya. “Using Machine Learning Algorithms to Identify Undervalued Baseball Players”. Stanford University. http://cs229.stanford.edu/proj2016/report/Ishii-UsingMachineLearningAlgorithmsToIdentifyUndervaluedBaseballPlayers-report.pdf

[10]: Arbesman, Samuel and Strogatz, Steven H. “A Monte Carlo Approach to Joe DiMaggio and Streaks in Baseball”. Cornell University, https://arxiv.org/ftp/arxiv/papers/0807/0807.5082.pdf

[11]: Everman, Brad. “Analyzing Baseball Statistics Using Data Mining”. Texas A&M University. https://truculent.org/papers/DB%20Paper.pdf

[12]: Koseler, Kaan and Stephan, Mattewh. “Machine Learning Applications in Baseball: A Systematic Literature Review”. Applied Artificial Intelligence. 26 Feb. 2018, http://www.users.miamioh.edu/stephamd/papers/Baseball_Machine_Learning.pdf

[13]: Chen, Leland and He, Andrew. “Beating the MLB Moneyline”. Stanford University. http://cs229.stanford.edu/proj2010/ChenHe-BeatingTheMLBMoneyline.pdf

[14]: Cserepy, Nico, Ostrow, Robibie and Weems, Ben. “Predicting the Final Score of Major League Baseball Games”. Stanford University. http://cs229.stanford.edu/proj2015/113_report.pdf

[15]: Hamilton, Michael, et al. “Applying Machine Learning Techniques to Baseball Pitch Prediction”. Jan. 2014, https://www.researchgate.net/publication/326972628_Applying_machine_learning_techniques_to_baseball_pitch_prediction

[16]: Albert, Jim. “Streaky Hitting in Baseball”. Journal of Quantitative Analysis in Sports. 2018, https://www.stat.berkeley.edu/~aldous/157/Papers/albert_streaky.pdf

[17]: King, Walter. “zWins, an Alternative Calculation of Wins Above Replacement in Baseball”. MIT Sloan Sports Analytics 2018. 11-12 Mar. 2016, http://www.sloansportsconference.com/wp-content/uploads/2016/02/1525-zWins-an-Alternative-Calculation-of-Wins-Above-Replacement-in-Baseball.pdf

[18]: Haddad, Peter, Kobza, Jake and Peynetti, Bruno. “Beat the Streak”. Northwestern Univesity. http://peterhad313.github.io/beat_the_streak_site/