

|  |  |
| --- | --- |
| To: | Dr. David Allee |
| From: | Chris Guise |
| Subject: | **Final Project – EEE498/591 Python for Engineers, Spring 2018** |
| Date: | 28 April 2018 |

**“Diamondback”: Python project for MLB machine learning predictions**

https://github.com/ChrisGuise/diamondback.git

*Python 3.6 project for analyzing MLB team season-by-season, month-by-month statistics to predict desired outcomes (i.e. playoff appearance) based on publicly available baseball game stats/logs. Highlights:*

* Implements machine learning algorithms and PCA(principal component analysis) to identify important stats predicting a desired outcome for input .csv data file
* Logistic regression ML algorithm to calculate %chance of desired outcomes

The package has two scripts:

* ImportData.py: helper tool to crunch Retrosheet data dump. Outputs MLBdata.csv
* MLBpredict.py: Imports teamStats.csv (can add more info manually if needed), runs ML algorithm, generates output .csv files (pcaVectors.csv, predictions.csv)

**Overview/Background:**

What adds up to a playoff team? Baseball pennant races are great fun to watch and one of my favorite parts of the game. Frequently, the improbable happens (see 2007 D’backs season). Or even the impossible (see 2007 Rockies, winning 21 of 22 games and in turn defeating the D’backs in the NLCS on their way to the World Series.))

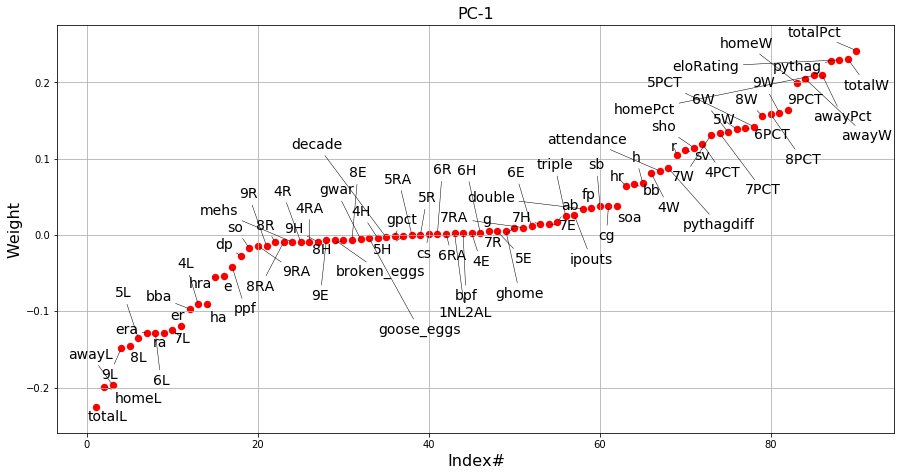
So…does a good bullpen pave a path to the playoffs? Or is it offense, or home runs, or starting pitching, or winning games decided by 1 run or less…or offense in September, strength of victories, run differential?…etc. Machine learning algorithms provide unique insight into these questions because of their transparency and ability to link together patterns in large multi-feature datasets.

**Executive Summary:**

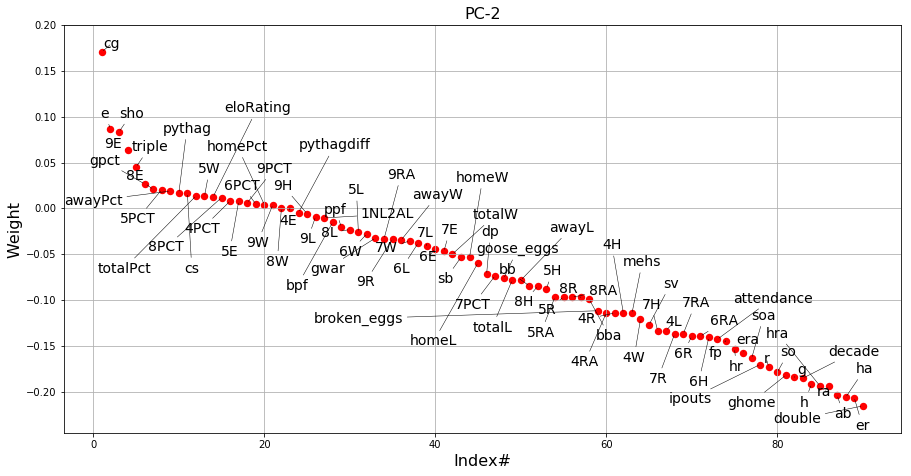
Raw data was obtained from several publicly available databases: Lahman Database, Retrosheet.org, and the FiveThirtyEight GitHub repository, and added manually to teamStats.csv. The sklearn.linear\_model.LogisticRegression ML method was used to calculate probabilities (lr.predict\_proba) as well as classify playoff vs. non playoff teams (lr.predict). Since there are only 30 x 53 = **1590** MLB baseball seasons by team between 1952-2015, “classical” machine learning techniques (along with careful feature selection) are likely a better fit vs. a deep learning / TensorFlow algorithm, given the small sample size.

**Results:** PCA (principal component analysis) creates new variables as the sum of eigenvectors that maximize variation from the mean within an N-dimensional space. Positive weights indicate positive correlation, negative weights indicate negative correlation. *Reversing the sign for all weights yields the same result since the eigenvector points in the same direction*[3].

For any #PCA components, component#1 alone predicts playoff teams with 90.0% accuracy:



* Total wins highest positive weight predicting playoff appearances (.232) and total losses goes the other way (-0.224). This vector passes the sanity check.
* Elo rating (0.220) and Pythag win% (0.221) also show strong weight/correlation
* Home and away record have roughly the same weight (0.203 vs. 0.207)
* Winning percentage in August & September is highly valued (0.161 Aug, 0.165 Sep) but winning percentage in April is less valued (0.123). This confirms observations from [2].
* Saves (0.128) correlate better to team playoff appearances than raw count of successful relief innings/ “goose eggs”[1] (-0.011).

2nd(V2) and 3rd PCA components improve accuracy to 90.7% and 91.3% respectively: 

* Note #complete games (0.173), #shutouts (0.091): great starting pitching makes playoffs
* PC2 shows correlation between low #earned runs (-0.197), low #hits allowed, HR allowed (-0.207, -0.191)…and low attendance (-0.135), poor offense (hits -0.176), and decade ID (-0.176). So especially in the “good ol’ days”, teams with great pitching but poor offenses playing in empty stadiums have a good chance of making the playoffs.

By training using the PCA features, the LR machine learning algorithm in turn determines whether the new “features” generated by PCA are useful vs. the desired outcome (in this case playoffs). Given better statistics, prediction accuracy may improve and other paths to the playoffs also found. More information is available in the commented code. Enjoy!

**References:**

1. Silver, Nate. (2017, April). “The Save Ruined Relief Pitching. The Goose Egg Can Fix It.” *FiveThirtyEight* [Online]. Available: <https://github.com/fivethirtyeight/data/tree/master/goose>
2. “When Do the Standings Matter?” *FanGraphs Community Research* [Online]. Available: <https://www.fangraphs.com/community/when-do-the-standings-matter/>
3. “Interpreting positive and negative signs of the elements of PCA eigenvectors.”. (2012) StackExchange [Online]. Available <https://stats.stackexchange.com/questions/26352/interpreting-positive-and-negative-signs-of-the-elements-of-pca-eigenvectors>