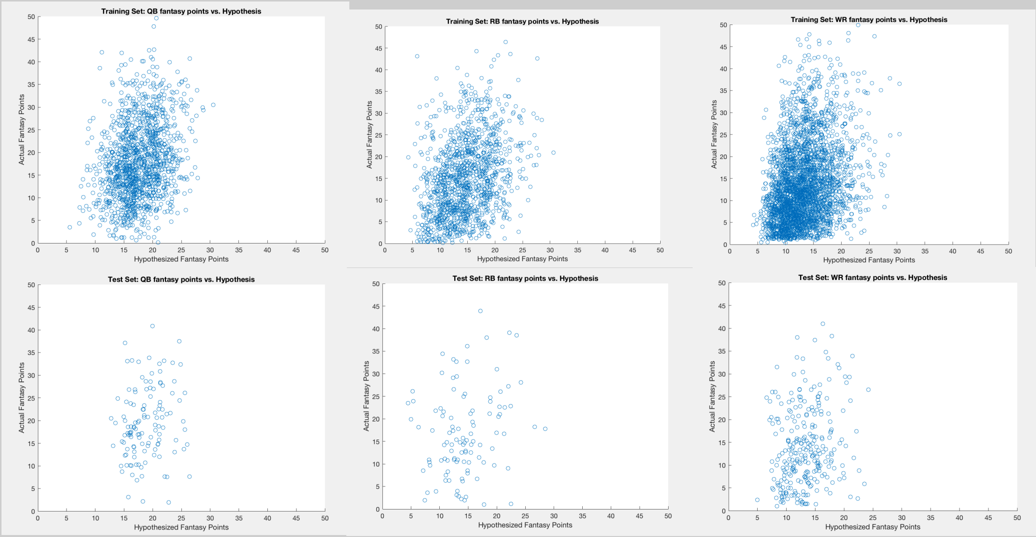
Beating Daily Fantasy Football

Matthew Fox, CS229, Stanford University

Abstract

Daily fantasy football provides an opportunity for fans across the country to build a lineup given a set budget and costs per player that the fan believes will have a high point total calculated from the player's stats in a given weekend. Companies such as DraftKings and FanDuel provide a marketplace for fans to create these lineups and play against each other. The premise of this project is that a machine learning algorithm can learn to create higher scoring daily fantasy football lineups more consistently than the average fan and consistently enough to overcome the rake that DraftKings takes for sponsoring the competition. In order to create these predictions, I've modeled each skill position player's expected points in a given weekend using separate models trained from a database of previous performances. With these predictions, I created a constraint-solving algorithm to pick numerous advantageous lineups that are beneath the overall budget and greater than the expected average for a given competition.

Predictions vs. Actual Fantasy Points



Data Collection & Feature Manipulation

I built my linear regression models upon nflgame, an open-sourced python library which supports both real-time data gathering and historical game data dating back to 2009 [4]. This package comes with all data up until the current week downloaded alongside the package, so each query is much shorter, only accessing local data unless you are depending on real-time results. The downside to nflgame is that some of the features that I had hoped to use are not available or easy to impute from historical data. Below are the features which I used to create the initial linear regression algorithms for each skill position:

QB	RB	WR/TE
Passing yards	Rushing Yards	Receiving Yards
Passing TDs	Rushing TDs	Receiving TDs
Interceptions	Rushing Attempts	Receptions
Passing Attempts	Receiving Yards	
Completions	Receiving TDs	
	Receptions	

I included features for sliding windows of 1, 3, and 5 games to give the algorithm flexibility for bias towards recent results or consistent results. Another interesting feature manipulation which I had to do was to account for bye weeks and injuries, which was done by limiting the number of zero fantasy score data points. Finally, I initially ran linear regression with an intercept term but decided to remove it as it appeared that the algorithm was biasing towards a high intercept and negative theta values which didn't make sense for the problem statement.

Results & Future Work

- In the above results, the ideal graphs would look like an x=y graph. These linear regressions clearly have a positive correlation with the actual fantasy results, but the correlation isn't as strong as I would like. Due to this, I've begun collecting a new set of features including past week actual fantasy point totals. Also, I've modified my data collection algorithms to move bye weeks from the sliding windows of past week results in order to get more consistent data. These new features should prove to strengthen the correlation between hypothesized and actual fantasy points.
- With that said, I've also built a constraint solving algorithm which randomly creates lineups and decides to enter the lineups based on meeting the budget constraint and predicting a score that outperforms the tournaments expected payout floor. Using this algorithm, I've received preliminary results across multiple recent weeks and hundreds of submitted lineups that I am averaging between 15 and 20 percent returns on investment. Although the algorithm hasn't been tested on enough data for me to call these trusted, reproducible results, they certainly motivate continued work on this algorithm and on gathering more testing data.
- Future work on this problem includes modeling defense fantasy scoring, including the opposing defense/offense in the feature set, collecting more data in order to expand the feature set into higher granularity, and trying to use kernels or SVMs to see if another machine learning algorithm could be more successful in predicting fantasy points.

Related Work

[1] http://www.pro-football-reference.com/
[2] http://datashoptalk.com/double-yo-money/
[3] http://cs229.stanford.edu/proj2015/104_report.pdf
[4] http://ttic.uchicago.edu/~kgimpel/papers/machine-learning-project-2006.pdf
[5] http://pjoos.github.io/2016/05/23/nfl_1.html
[6] http://www.cs.cornell.edu/courses/cs6780/2010fa/projects/warner_cs6780.pdf
[7] https://www.credera.com/blog/business-intelligence/using-machine-learning-predict-nfl-games/
[8] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.8415&rep=rep1&type=pdf
[9] https://gigaom.com/2015/02/12/heres-more-evidence-that-sports-is-a-goldmine-for-machine-learning/