Finding the Optimal Fantasy Football Team

Paul Steenkiste
pws@stanford.edu
December 11, 2015

I. Introduction

Fantasy football, just like the game of football itself, as grown from a benign neighborhood pastime into an immensely profitable corporate industry. Over the past decade, the number of individuals who play in a fantasy football league has grown by 450% to a total of 56.8 million people.[1] Through their purchase of website memberships, league entrance fees, and draft 'cheat sheets,' fantasy football has become a $27 billion dollar market. To put this into perspective, the average valuation of an NFL team is a mere $1.97 billion (while the team with the highest net worth is the Dallas Cowboys, at 'just' $4 billion).[2]

However, outside the television pundits and online soothsayers, there has been relatively little formal work done on how to predict the fantasy value of a given player. For this project, I used various machine learning tools to try to do just that. I focused on the largest-growing sector of the industry: daily fantasy leagues, where participants pay an entry fee and are given a certain amount of "fantasy dollars" with which to purchase a team, with one player for every position. Each player's price is determined based on how well the league projects him to perform that week.

Because of the lack of rigorous examination of the expected payout of any given player, there are sure to be mispriced players whose hidden value can be exploited by a fantasy competitor. I aim with this project to help the fantasy sports industry move closer to an equilibrium that would be predicted by the Efficient Market Hypothesis, where asset prices fully reflect all available information.

I used three models: Linear Regression, Random Forests, and Multivariate Adaptive Regression Splines. For each model, I used as input every player's statistics in the previous weeks of the season, as well as the team he was playing in each week. I then predicted how each player would do against his specific opponent in each of the relevant fantasy categories (passing yards, rushing touchdowns, etc.) in the coming week.

II. Related Work

As previously stated, very few people have attempted to use machine learning to predict fantasy football outcomes (or, if they have, they are using their findings to take advantage of the information asymmetries in this infant market). In fact, I was only able to find a single research paper on the topic: Roman Lutz of the University of Massachusetts Amherst used Support Vector Regression to try to predict the fantasy scores of NFL quarterbacks, with some promising results.[3] Seeing that SVMs were already attempted, I decided to build a basic linear regression model to see how it performs, and then run diagnostics on its performance to see where it would be most fruitful to go next.

III. Dataset and Features

I gathered the weekly player statistics for each of the first 11 weeks of this NFL season from NFL.com.[4] Considering that there are 32 teams playing almost every week, each team having around nine players making offensive statistical contributions on any given Sunday, this came out to around 3,200 data points.

For every week from NFL Week 4 through NFL Week 11, I trained each model using the data from all the previous weeks, and tested each model on the current week. I decided to start training after three weeks of data had been logged because a player's statistics from just one or two games would probably not be enough to predict his future performance. The X values for each data point were the player and opposing team for the given contest, and the Y values were the resulting fantasy statistics for that player.

To represent each player in vector form, I simply constructed a binary dimension for each player. Because there were 531 total players in the data set, this means that I created 531 dimensions. For any given data point of a specific player in a specific game, each of these added dimensions contained a '0' except for the one corresponding to that player. I did the same process to represent the opposing team the player was going against in that game. Because there are 32 teams in the NFL, I added another 32 dimensions, each of which contained a '0' except for the one corresponding to the correct opposing team. In total, therefore, each X value was a vector of 563 binary values (only two of which were '1').

The fantasy leagues I was evaluating used nine statistical categories to give each player a fantasy score (See Table 1). For each data point, the player's statistical performance in each category represented the Y value.

Table 1: Fantasy Scoring System

Statistical Category	Fantasy Points
Passing Yards	0.04
Passing Touchdowns	4
Interceptions	-1
Rushing Yards	0.1
Rushing Touchdowns	6
Fumbles	-2
Receptions	0.5
Receiving Yards	0.1
Receiving Touchdowns	6

When discussing the error of my models in this paper, I will be referring to the sum of *normalized* squared error as given by following equation:

$$\sum_{i \in test\ set} \left(\frac{\hat{y}_i - y_i}{\hat{\sigma}_y}\right)^2$$

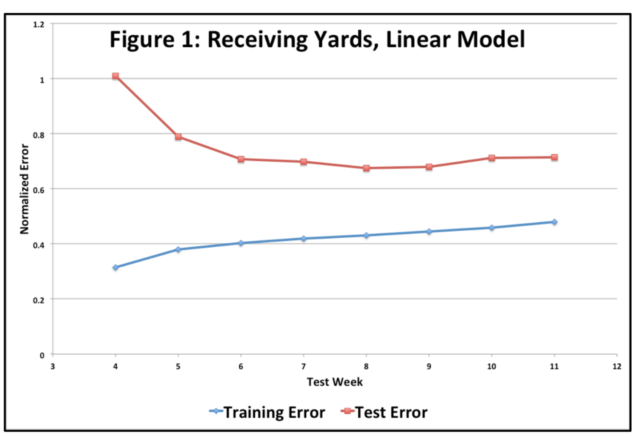
Normalizing the squared error by the estimate of the population's standard deviation is necessitated by the differing units of the statistical categories: if my model is able to predict a running back's rush yards to within ten yards, that should be considered a resounding success. However, if my model predicts that he will get ten more touchdowns than he eventually does, that would be a grave error indeed. I owe the idea to normalize this way to Daniel O'Neel and Reed Johnson's 2014 CS229 paper.[5]

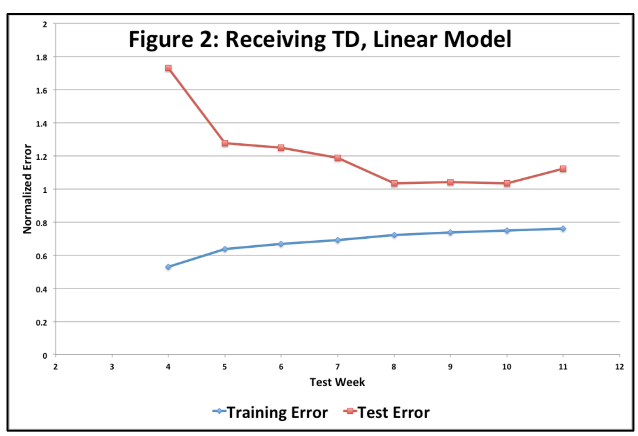
IV. Methods

Model 1: Linear Regression

To begin my examination of fantasy football predictions, I decided to run a basic linear model, using MATLAB's 'fitlm' function. Training and testing this model for every week, I generated learning curves for each statistical category (See Figures 1 and 2 for examples).

Each of the learning curves had the same general shape: the model would steadily improve until roughly midseason, and then the slope of the test error would flatten out, with a significant gap between the training and test errors. This implied that the model suffered from a variance problem more than a bias problem.





A common remedy to high variance is simply to collect more data points; however, since I had collected every possible point from the current NFL season, my options were limited. I could have collected data from previous years to try to use a player's performances from earlier in his career as predictors. However, considering the constantly-shifting landscape of personnel and scheme in the NFL, as well as the fact that some of the top performers in the league are rookies who do not have previous seasons to analyze, this strategy seemed unlikely to bear fruit.

Model 2: Random Forests

Random Forests (RFs) are a regression method that builds many regression trees (with no pruning) from the inputted data, and then outputs the mean prediction of all of the individual trees. In order to bifurcate the trees, RFs form an estimation of what variables are important to the regression, and thus they work well with data sets with large feature sets (like my own).[6] Further, one of the reasons RFs were developed was to correct for decision trees' proclivity to overfit the data; thus, it seemed a perfect tool to combat the variance of my linear model.[7]

To implement RFs, I used R's 'randomForest' package. As will be seen, RFs did turn out to be an effective tool in predicting scores, especially towards the beginning of the season when other models were still suffering from very high test error.

Model 3: Multivariate Adaptive Regression Splines

Multivariate Adaptive Regression Splines (MARS) is a nonparametric regression method that makes no immediate assumptions about the relationship between the data. Using spline functions (piecewise-polynomials that allow there to be "kink" points in the regression lines), MARS automatically models nonlinearities and interactions between the variables.[8] As I was unsure whether there were any such hidden relationships in my dataset, MARS proved immensely useful. Further, because it partitions the input space into multiple regions, it is also particularly suitable for data with large feature sets (again, like my own).[9]

To implement MARS, I used R's 'earth' package. Because MARS allows arbitrarily many degrees of interaction between the variables, I needed to determine up to what degree the model would be effective. MARS, as a nonparametric model with a lot of flexibility, tends to overfit the data if the degree is too high, resulting in artificially low training error but inflated test error. The 'earth' package in R, however, uses a pruning technique to limit the allowed complexity of the model; therefore, if the degree is set too high, the package simply ignores the higher interaction terms. In our case, the model minimized test error at Degree 3, and so for all subsequent degrees the errors were precisely the same (see Figure 3).

Figure 3: Week 11 Passing Yards Error, MARS

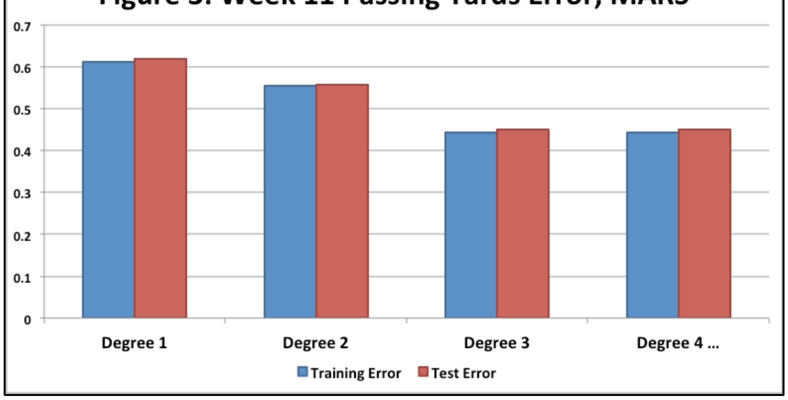
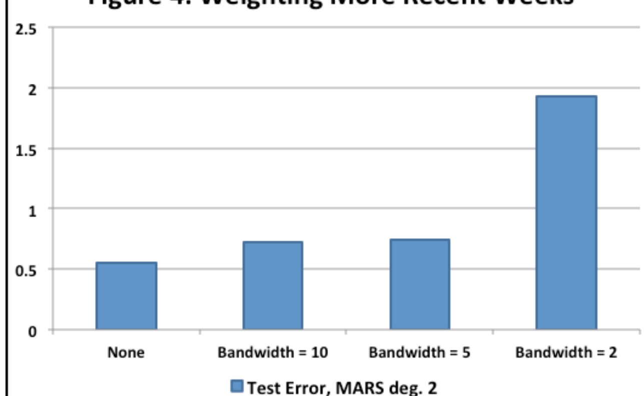


Figure 4: Weighting More Recent Weeks



Model 3.i: Weighted MARS

Finally, it occurred to me that the model might be improved if recent weeks were weighted more heavily than performances from earlier in the year. Conceivably, a player could have a "trajectory" through the season; perhaps he is an aging player whose physical abilities are declining, or a rookie who is learning the ropes and getting steadily better.

Using the conventional "Gaussian" weighting formula with various bandwidths, I ran MARS and found that the test errors steadily increased as the bandwidth got smaller, with the lowest error being that of the original unweighted model (see Figure 4). Therefore, more recent games have no more predictive power than earlier games, and the model was needlessly throwing away information by using different weights.

Weight Formula:

$$w^{(i)} = \exp\left(-\frac{\left(x^{(i)} - x\right)^2}{2\tau^2}\right)$$

where $w^{(i)}$ = weight for data point $\vec{x}^{(i)}$
$x^{(i)}$ = week of data point $\vec{x}^{(i)}$
x = week we are trying to predict
τ = bandwidth parameter

V. Results

Figures 5 and 6 show how well the models performed relative to each other in two statistical categories. In most categories, the MARS models consistently performed the best in terms of normalized test error.

Figure 5: Pass Yards Test Error

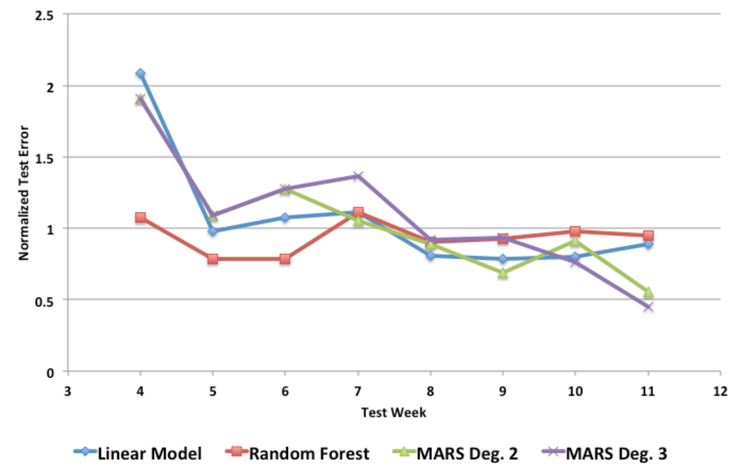
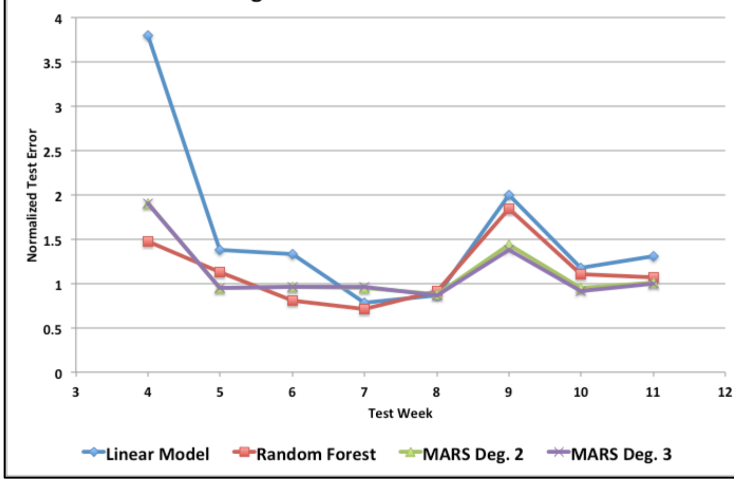


Figure 6: Pass TD Test Error



There were a few categories that were inherently more difficult to predict across all models, most notably fumbles and interceptions (see Figure 7). This makes some intuitive sense: while statistics such as total yards are accumulated over the course of an entire game (and thus not as susceptible to random chance), a game's solitary interception can be caused by the slightest slip of the finger by the quarterback, and the one fumble in a running back's season occurs because the defender's helmet hit the ball in just the right location. Thus, turnovers are immensely difficult (if not impossible) to predict.

Figure 7: Interceptions, Linear Model

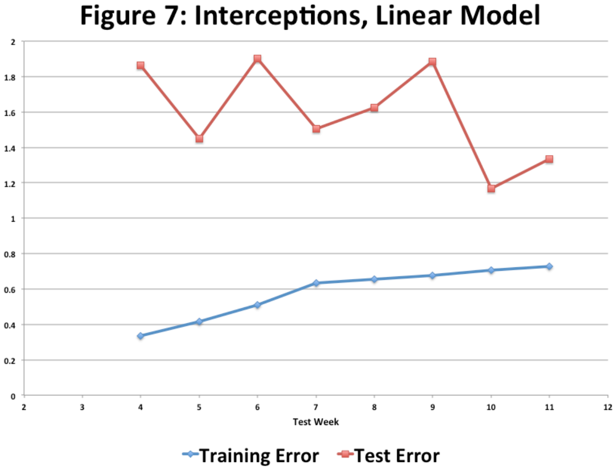


Table 2: Projected Quarterback Rankings, Week 11

(Actual rank in parentheses)

	Our Models			Professional Prognosticators	
Proj. Rank	Linear Model	MARS (deg. 2)	Random Forest	Bleacher Report	Fantasy Pros
1	Jay Cutler (15)	Blaine Gabbert (7)	Andy Dalton (4)	Tom Brady (10)	Tom Brady (10)
2	Blaine Gabbert (7)	Jay Cutler (15)	T. Bridgewater (6)	Cam Newton (1)	Cam Newton (1)
3	Jameis Winston (2)	Jameis Winston (2)	Matt Hassleback (20)	Aaron Rodgers (12)	Philip Rivers (25)
4	Cam Newton (1)	T. Bridgewater (6)	Blaine Gabbert (7)	Derek Carr (24)	Carson Palmer (3)
5	T. Bridgewater (6)	Andy Dalton (4)	Matthew Stafford (8)	Philip Rivers (25)	Aaron Rodgers (12)
6	Andy Dalton (4)	Cam Newton (1)	Ryan Fitzpatrick (23)	–	Blake Bortles (14)
7	Joe Flacco (11)	Ryan Fitzpatrick (23)	Jay Cutler (15)	–	Derek Carr (24)

To see how my models performed relative to real-life sports columnists and commentators, I ran the quarterback data through all three models and predicted who the top seven fantasy options would be for Week 11. I then compared my rankings to those of two professional prognosticators (see Table 2).[10, 11]

As one can see, my rankings did very well compared to those of the analysts. All three models correctly chose four or five of the players who did actually end up ranked in the top seven (highlighted in green above), with relatively few gross over-appreciations (defined as players who actually ranked twentieth or worse, highlighted in red above). In contrast, both professional analysts grossly overvalued two players: Derek Carr (24^{th}) and Philip Rivers (25^{th}). These missteps are especially embarrassing considering that there were only 25 quarterbacks in the model.

One of the principal reason my models outperformed the 'experts' is that the latter base their decisions largely on the reputation of quarterbacks. Notice that both Tom Brady and Aaron Rodgers were highly ranked by the two experts. Brady and Rodgers are two of the most renowned and prominent players in the league; they have had a lot of success in years past (and, indeed in some weeks this year), and so many simply assume they will always play well going forward. However, while neither of them performed horribly, they both only ended up in the middle of the pack. My model, instead of relying on name recognition, took all the available information – who they were playing, how they did against specific defenses in the past – and correctly ranked them outside the top seven.

There are some things even my rankings could not predict, however: notice that Jay Cutler, who eventually earned mediocre statistics that week, is ranked in the top seven by all my models. Cutler had been having a very strong season up to that point, but since that week he has been playing relatively poorly. There was no indication in the previous statistics that this would happen, so my model could not have known. Football is a human game, after all, and some things simply cannot be predicted.

VI. Conclusion

My models can already be used by fantasy competitors to purchase a team that will be successful with high probability. One must simply look up the team matchups for the week, input the binary data into the models, and receive back fantasy score projections. From there, it is a relatively simply linear optimization problem to discover the 'best' team you can buy with your fantasy budget constraint:

$$\max FantasyPoints$$

$$s.t. \sum_{Purchased\ Players} Cost\ of\ Player \leq BudgetConstraint$$

$$\sum Purchased\ Players = 1 \quad \forall Positions$$

All of the above can be represented in matrix form, and inputted into any linear optimization software.

Going forward, however, my models can definitely be improved. There are many more input variables that can be tested for predictive power. For instance, how experienced is the player? If he is a rookie, he is probably not going to be able to have the immediate statistical success that a battle-hardened veteran might. Is the player recovering from a recent injury? This season, a very good quarterback named Tony Romo was injured and could not play for a few weeks. When he returned, many expected him to return to his winning ways. Romo played very poorly at first, however, taking some time to get back into form. Perhaps if injuries had been incorporated into my models, this could have been predicted. Further, a player's injuries can have implications for his teammates as well. This season another great quarterback, Ben Roethlisberger of the Steelers, also had to take a few weeks off because of an injury. During this time, one of the best wide receivers in the NFL, Antonio Brown, also on the Steelers, had some relatively poor statistical Sundays because he was forced to play catch with an inferior backup quarterback. If my model had somehow incorporated teammates' injuries, perhaps this, too, could have been predicted.

Overall, however, I am happy with the performance of my models to this point, as they have already bested the predictions of people who get paid to predict football statistics for a living. Perhaps my dream of an Efficient Fantasy Football Market, where fantasy values fully reflect all available information and player futures are traded on the margin, is not too far off.

References

1. Affleck, John. "What's Behind Fantasy Football's Surprising Popularity." *Fortune*. 12 Sept. 2015. http://fortune.com/2015/09/12/fantasy-football-growth/.
2. Ozanian, Mike. "The Most Valuable Teams In The NFL." *Forbes*. Forbes Magazine, 14 Sept. 2015. http://www.forbes.com/sites/mikeozanian/2015/09/14/the-most-valuable-teams-in-the-nfl/.
3. Lutz, Roman. "Fantasy Football Prediction." *arXiv preprint arXiv:1505.06918* (2015).
4. http://www.nfl.com/stats/weeklyleaders
5. O'Neel, Daniel and Reed Johnson. "NFL Defensive Performance Analysis," Fall 2014 CS229 Project, http://cs229.stanford.edu/proj2014/Daniel%20ONeel,%20Reed%20Johnson,%20Defensive%20Unit%20Performance%20Analysis.pdf.
6. "Random Forests." *Berkeley Statistics Department*. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#overview.
7. Breiman, Leo. "Random Forests." *Machine Learning* 45.1 (October 2001): 5-32.
8. StatSoft. "Multivariate Adaptive Regression Splines (MARSplines)." *Multivariate Adaptive Regression Splines (MARSplines)*. Dell Software. https://www.statsoft.com/Textbook/Multivariate-Adaptive-Regression-Splines.
9. Friedman, Jerome H. "Multivariate Adaptive Regression Splines." *The Annals of Statistics* 19.1 (1991): 1-67.
10. Loechner, Tyler. "Fantasy Football Week 11 Preview and Predictions." *Bleacher Report*. 17 Nov. 2015. http://bleacherreport.com/articles/2590198-fantasy-football-week-11-preview-and-predictions/page/2.
11. Borgognoni, Kyle. "Fantasy Football: QB1 Primer Week 11." *FantasyPros*. 19 Nov. 2015. http://www.fantasypros.com/2015/11/fantasy-football-qb1-primer-week-11/.