Predicting Fantasy Football Performance with Machine Learning Techniques

Introduction and Background

Once a paper and pencil game played only by a few sports aficionados, the internet has helped transform fantasy sports into a \$1 billion dollar industry. Accounting for nearly 40% of this industry is football, with millions of casual fans playing in fantasy football leagues every year.

The basic premise of fantasy football is as follows. A fantasy football league, typically consisting of 8-10 competitors, holds a "draft" before every NFL season where each fantasy competitor has a limited number of virtual resources (usually a salary cap or a fixed number of draft picks) available to spend. Using these resources, each competitor selects a virtual team comprised of real NFL athletes. Fantasy competitors then face one another in heads-up games every week of the NFL season, with scoring in the fantasy games dictated by the statistical in-game performance (i.e. yards gained, touchdowns scored, etc.) of the NFL athletes in their actual games.

The major challenge of fantasy football is therefore to select players who provide good statistical performance relative to their price in the draft. As an avid fantasy football player, I decided to focus my final project on building statistical models to predict the NFL athletes who will score the most fantasy points in a given season.

Project Scope

In general, fantasy teams consist of at least one quarterback, two wide receivers, two running backs, a field goal kicker, and a tight end. To limit the scope of the project, *this project will generate pre-season predictions for running backs (RBs) only*. However, results from this project can be generalized to develop models for all other NFL positions as well.

Fantasy Point Rules for Running Backs

Fantasy point scoring for a running back in a given week is given by the following two simple rules:

A running back gets 1 fantasy point every time he gains 10 yards

A running back gets 6 fantasy points every time he scores a touchdown

First Crack at the Problem - Using Linear Regression

My first project goal was to get a very simple learning model up and running. Given that the number of fantasy points scored by a running back can be viewed as a continuous output, I decided to start with a simple linear regression model with only two features¹:

 $x_1 = \frac{\textit{Number of Yards Gained By Running Back in previous season}}{\textit{Number of Games played by Running Back in previous season}}$ $x_2 = \frac{\textit{Number of Touchdowns Scored By Running Back in previous season}}{\textit{Number of Games played by Running Back in previous season}}}$

¹ I chose to normalize my feature vectors by the number of games a running back played in a given season, to avoid penalizing running backs who missed games due to injury/suspension/contract disputes, etc.

The model therefore predicts fantasy point scoring for a running back solely on how many yards and touchdowns they had in the *previous* year. This is admittedly a simple choice of a feature vector, but since fantasy point scoring is exclusively dependent on scoring touchdowns and gaining yards, it makes sense to start with this choice of feature vector as a baseline.

Data Collection:

A training set was collected from the statistics of m= 34 running backs finishing with at least 70 fantasy points in both the 2007 and 2008 NFL seasons. The yardage and touchdown statistics to form the feature data x were collected from 2007, and the fantasy point totals for the target variable y were collected from $2008.^2$

Results of Linear Regression:

To test the regression model, I made predictions of how 32 running backs would perform in the 2010 NFL season, based on their performance in the 2009 NFL season. Figure 1 shows a learning curve for the regression algorithm. The error metric on the y axis is the average estimation error between predicted and actual running back performance, given in fantasy points/year. The figure shows that the estimation error stays roughly constant after m = 15, and that our average estimation error is slightly higher, but on par with the predictions of Mike Krueger, a human expert who makes fantasy predictions for fftoday.com.

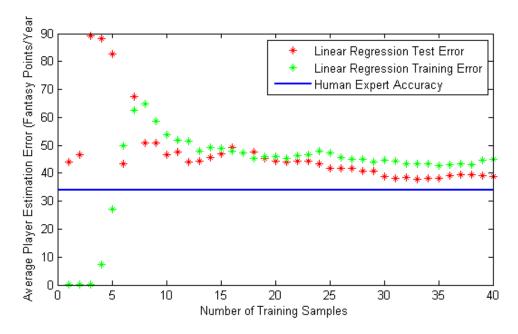


Figure 1: Learning curve for linear regression algorithm. Note that training and test error are similar as number of training samples increases.

² The 70 point cutoff for the training set was chosen to exclude running backs whose first season in the NFL was in 2010, as well as running backs that missed significant time due to injury in either season. Data was collected from fftoday.com

While a reasonable metric to evaluate a learning curve, "average prediction error" as defined above is not the best metric for comparing two prediction methods, since winning in fantasy football is about relative performance between running backs. A better way is to evaluate the algorithm is to use the numerical predictions to create a ranked list of running backs for the upcoming season, and then see how these picks actually end up performing in 2010. This is shown in Table 1.

Predicting the Top 10 Running Backs of 2010								
Linear Regression	Predicted Points	Human Expert (Mike Krueger)	Predicted Points	Actual 2010 Rankings	Actual Points			
Chris Johnson	242	Adrian Peterson	283	Arian Foster	329			
Adrian Peterson	241	Chris Johnson	277	Peyton Hillis	243			
Maurice Jones	233	Maurice Jones	270	Adrian Peterson	231			
Drew		Drew						
Frank Gore	222	Ray Rice	246	Jamaal Charles	231			
Ray Rice	221	Frank Gore	224	Chris Johnson	232			
Thomas Jones	217	Ryan Mathews	220	Darren McFadden	226			
Steven Jackson	215	Rashard	220 Rashard		222			
		Mendenhall		Mendenhall				
Cedric Benson	213	Steven Jackson	215	LeSean McCoy	222			
Michael Turner	206	Michael Turner	211	Michael Turner	217			
Ricky Williams	202	DeAngelo Williams	210	Matt Forte	215			

Table 1: Running back predictions compared to actual results. First column is from my linear regression algorithm, second column is from a human expert, third column is actual results. Values in parenthesis represent predicted/actual points scored. Rankings accurate to within five positions are shown in green. Questionable picks are shown in red.

The difficulty of predicting fantasy performance is immediately apparent. Very few people predicted the explosive emergence of Jamaal Charles, LeSean Mccoy, andAdrian Foster, who were two young newcomers to the NFL in 2010. Similarly, the injury of Maurice Jones-Drew, one of the NFLs most consistent running backs, shook up the final season rankings further. A second observation is the relative similarity between Mike Krueger's predictions and the predictions from linear regression. The two sets of predictions share seven common players, each ranked within 1-2 spots of one another.

Another interesting observation is the regression algorithm's high ranking of Thomas Jones and Ricky Williams. While both athletes had solid 2009 seasons, both players were moved to backup roles before the 2010 season as they competed for playing time with younger running backs on their teams. Most fantasy football experts, Mike Krueger included, therefore had these two ranked well outside the top 30, as it was unlikely they would repeat their 2009 performance. Without a way to capture this preseason information, the algorithm as presented is unable to recognize the risk associated with these two players.

A Second Attempt at the Problem – Using a Clustering Algorithm

An alternate approach to predicting good fantasy football players is to group NFL running backs into several clusters, based on a variety of features such as number of games played, number of rushing attempts, rushing yards, touchdowns, and total fantasy points scored. Player predictions are then made by first classifying running backs into their corresponding group, and then applying a regression model unique to that group.

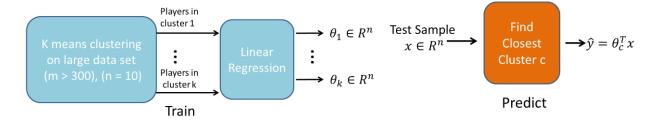


Figure 2: Combination of clustering and linear regression algorithm used to make predictions

The idea behind this method is that there may be several fundamental types of running backs in the NFL. In this case, it's possible to get more accurate predictions by having a different set of linear regression coefficients for each type of player. For example, players who were injured in one season will have an artificially low number of fantasy points scored that year, and will often see a dramatic increase in fantasy points the next year simply by being healthy. This cluster might therefore have relatively larger regression coefficients compared to a cluster of players who stayed healthy.

To perform the k-means clustering, I gathered a larger dataset of training data, encompassing the statistical performance of m = 292 running backs from 2006 to 2008. After experimenting with a number of feature combinations, I found it best to cluster the running backs using only three features: number of games played, total yards per game, and total touchdowns per game.

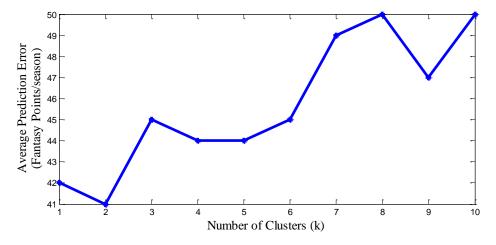


Figure 3: Player prediction error as a function of clusters used

To determine the number of clusters to use, I calculated the average prediction error (the same metric used for linear regression) for a variety of k (see Figure 3).

I found that in terms of this metric, the number of clusters to use wasn't immediately obvious, as the prediction error hovered around 42-45 points per year for k=1 to 6. However, I found that as the number of clusters increased beyond six, the clustering algorithm tended to get stuck in local minima and came up with increasingly erroneous predictions. In terms of qualitative performance, I found that the machine learning algorithm came up with the most reasonable picks at k=3 or 4.

	Games	Yards /Game	TD /Game	% Training Data
Cluster 1	14.4	95.2	0.43	17.4
Cluster 2	15.5	26.6	0.14	17.1
Cluster 3	11.7	38.4	0.21	34.5
Cluster 4	5.0	36.4	0.07	31.0

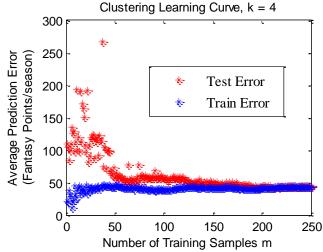


Table 2: Cluster Centroids Found for k = 4

Table 2 shows the cluster centroids for k = 4. The algorithm splits about one third of the data into Cluster 4, who appear to be players dealing with injury in 2009. Another half of the players are split into Clusters 2 and 3, low performing clusters typical of average NFL running backs. On the other hand, Cluster 1 represents the small but very important number of elite running backs in the NFL. A learning curve is also plotted for k = 4 as well, showing convergence after about m = 150.

Predicting the Top 10 Running Backs of 2010								
Clustering	Pred. Pts.	Human Expert	Pred. Pts.	Actual 2010	Actual Pts.			
Chris Johnson	251	Adrian Peterson	283	Arian Foster	329			
Adrian Peterson	232	Chris Johnson	277	Peyton Hillis	243			
MJ. Drew	215	M.J. Drew	270	Adrian Peterson	231			
Frank Gore	215	Ray Rice	246	Jamaal Charles	231			
Michael Turner	221	Frank Gore	224	Chris Johnson	232			
Thomas Jones	203	Ryan Mathews	220	Darren McFadden	226			
Joseph Addai	184	R.Mendenhall	220	R.Mendenhall	222			
Ricky Williams	177	Steven Jackson	215	LeSean McCoy	222			
L. Tomlinson	173	Michael Turner	211	Michael Turner	217			
D.Williams	169	D. Williams	210	Matt Forte	215			

Table 3: Predictions for 2010.

Table 3 shows the top ten projected picks for 2010 using the clustering algorithm. The clustering algorithm makes predictions similar to the original linear regression algorithm, although we have now another questionable top ten pick in a rather old LaDainian Tomlinson.

Conclusion

Given the large number of unpredictable factors, it is very difficult for both humans and computers to pick who the best NFL running backs in a given season will be. The first linear regression algorithm presented is very easy to implement and gives results on par with human experts, but needs additional features accounting for offseason injuries, increasing age, and loss of playing time due to new players entering the league. Clustering offers an interesting way to group players with similar historical performance, but still needs these difficult-to-collect features. If I were to expand upon this project, adding playing time and age information would be a top priority. Additionally, I might also make each training sample contain feature data from the past several seasons, instead of just the prior season.