Predicting Running Back Performance in Daily Fantasy Football with a Basic Tool to Receive User Input

Venkat Motupalli, GIT: vrmotupalli, vm1218@nyu.edu

Abstract

Daily Fantasy Sports (DFS) are now a multi billion dollar business, with records for participants and money deposited set again on 11 October 2015.[1] While contests exist for all major professional sports, and many minor ones, fantasy football remains the most popular first timer game, and the most inclusive.[2]

This project seeks to build on previously successful work by the author to add variable parsimony to a logit model of football running back (RB) performance in DFS, and develop a rudimentary script to accept user input and return a likelihood of performance compared to a threshold. Ideally, this project will result in similar or greater success with less required data, and deliver the foundations of a proof of concept product to be used in DFS.

Introduction

Optimal fantasy football lineups remain challenging to predict statistically. Elite business schools – including Stern – regularly publish lineup optimization models with varied success. Columns abound on the internet, most drawing from respected data aggregators. The most commercially effective of these are Matthew Berry's Love/Hate for ESPN, and Jaime Eisenberg's Start 'em Sit 'em for CBS. Both follow an up down format, projecting a variety of players that they think will over or underperform that week.

The up down format is more likely to be useful than canned lineups for the casual or amateur DFS player. On Twitter, Matthew Berry himself responded that he – and by corollary ESPN – would immediately integrate effective statistical analysis for DFS into his site if it was provided.[3]

In previous work, a model was developed that described the likelihood of a fantasy football running back (RB) to over or underperform an analytically determined threshold, and provide recommendations for Daily Fantasy Sports use. While this model was both successful and useful, it required a large amount of curated inputs, some of which had apparently negligible marginal effects.

While manual data handling is largely necessary because of both availability and intangible football considerations, there is significant potential to introduce variable parsimony, reducing the data preparation requirement of the model. Further, given the proven success and the fact that the model is built in a freeware language (Python), this project will explore adding interactivity – that is, user input –

---

[1] Roberts, D. (2015, October 19). DraftKings, FanDuel Entries Down This Weekend. Retrieved October 19, 2015, from http://fortune.com/2015/10/19/draftkings-fanduel-first-down-sunday/

[2] What's behind fantasy football's surprising popularity. (2015, September 12). Retrieved October 19, 2015, from http://fortune.com/2015/09/12/fantasy-football-growth/

[3] Berry, M. (2015, June 30). 100 facts for the 2015 season. Retrieved October 19, 2015, from http://espn.go.com/fantasy/football/story/_/id/13101710/100-facts-2015-season-fantasy-football

in order to evaluate specific players. Specificity will broaden the applications of this model to all fantasy football competitions, not just DFS.

Data

NFL.com publishes all accepted running back metrics per game, per season. These metrics are split into separate rushing and passing datasets. To train the original model, the last three years of each dataset were into one on the players' names. For this project, that dataset was truncated in order to have an easily shareable dataset for demonstration purposes. This dataset informs nine metric variables affecting performance associated with each active RB in the NFL.

CUSP provides a city to city airport connection database that was filtered to isolate football city connections in the US. The connections to the two London games were discounted. The resulting dataset provided just over 700 connections between the 31 football cities, which were then merged to the RB metric dataset on the players' teams.

Methodology

This project uses DraftKings.com scoring as representative of the general DFS offerings. RBs were chosen as the experimental position because they have the ability to score points in multiple categories, and as such, have an outsized effect on lineups per fantasy dollar.

Accepting that it is not possible to easily predict single players' performance in order to optimize an entire lineup mathematically, it is certainly possible to develop lists of players who are statistically likely to outperform a threshold.

The logit model developed previously used all available NFL.com RB metrics to return the likelihood of a running back to outperform 12 fantasy points with a 77% and 90% success rate predicting outperforming and underperforming RBs respectively. In this project, variables with apparently low marginal effects, shown below, were removed both sequentially and simultaneously to test the outcome.

```
In [3]: RBlogitmod = smf.logit('PERF ~ PLAYS + RUN + RYD + RTD + TARGETS + REC + PYDS + PTD + FUM + Connections', data = data).fit()
        print
        print RBlogitmod.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.194123
         Iterations 8

                           Logit Regression Results
==============================================================================
Dep. Variable:                   PERF   No. Observations:                  147
Model:                          Logit   Df Residuals:                      136
Method:                           MLE   Df Model:                           10
Date:                Sun, 13 Dec 2015   Pseudo R-squ.:                  0.5401
Time:                        15:18:03   Log-Likelihood:                -28.536
converged:                       True   LL-Null:                       -62.051
                                        LLR p-value:                 1.654e-10
===============================================================================
                  coef    std err          z      P>|z|      [95.0% Conf. Int.]
-------------------------------------------------------------------------------
Intercept      -4.0618      1.462     -2.779      0.005        -6.926    -1.197
PLAYS           2.0577      1.774      1.160      0.246        -1.419     5.534
RUN            -2.0601      1.777     -1.159      0.246        -5.543     1.423
RYD             0.0070      0.007      1.033      0.302        -0.006     0.020
RTD             0.6699      0.295      2.272      0.023         0.092     1.248
TARGETS        -2.0937      1.775     -1.180      0.238        -5.573     1.385
REC            -0.0585      0.148     -0.395      0.693        -0.349     0.232
PYDS            0.0121      0.011      1.127      0.260        -0.009     0.033
PTD             1.2567      0.644      1.950      0.051        -0.006     2.520
FUM            -1.3828      0.597     -2.316      0.021        -2.553    -0.213
Connections    -0.0496      0.053     -0.944      0.345        -0.153     0.053
===============================================================================
```

Finally, significant code was added around the model in the script to accept user input for specific players, creating a tool that can be used in all fantasy formats, rather than just DFS.

Conclusions

Removing variables sequentially necessarily changed the marginal effects of the remaining decision variables, occasionally on an order of magnitude, but those effects are judged to be proportional and not descriptive. In general, removing all four variables with relatively low marginal effects resulted in a similarly effective model.

With all four variables identified above removed, and running the model across 32 starting RBs in DK for football Week 10, the model maintains approximately 74% and 90% success predicting outperforming RBs and underperforming RBs respectively. It rejects 25 of the top 40 runners – or the RBs that fantasy players are likely to focus on, delivering a potential 62% increase in efficiency if each RB is considered a unit of time in the lineup building process.

This is significant because the model with less variables negates the need for two of the four datasets used – receiving statistics and airport connections – making it much more user friendly week by week. This also identifies the potential to consider summed statistics from the two remaining datasets, which is discussed below.

The attached iPython notebook details both models, and ends with code prompting user input for specific players. The user input and output is displayed below.

```
Number of plays? 199
Number of running plays? 180
Number of rushing touchdowns? 4
Number of targets? 19
Number of receiving touchdowns? 0
Number of fumbles? 1
```

```
# 199, 180, 4, 19, 0, 1 Adrian Peterson
e = np.exp(RBlogitmod.params['Intercept'] + RBlogitmod.params['PLAYS']*PLAYS +
                RBlogitmod.params['RUN']*RUN + RBlogitmod.params['RTD']*RTD +
                RBlogitmod.params['TARGETS']*TARGETS + RBlogitmod.params['PTD']*PTD + RBlogitmod.params['FUM'] * FUM)
print(e / (1 + e))
```

```
0.634151719877
```

As shown, Adrian Peterson is correctly predicted to be more likely to outperform 12 fantasy points in football week 10, 2015. This allows the model to not only be practically useful as a list generator for DFS, but a possession evaluator for all other types of fantasy formats.

Future Work

Given that two of the four previously used datasets are now identified as practically extraneous, it is possible to examine sums of the remaining metrics as viable decision variables. If certain combinations of summed metrics are viable, it makes automating data collection possible, as those combinations are available simultaneously in useful formats.

Further, given that somewhat non intuitive combinations of metrics seem to be effective, there is value to this kind of granular variable selection while evaluating other offensive positions in DFS.