Machine Learning - Fantasy Football Predictor

Christopher Beauchene
Northwestern University
Computer Engineering and Economics

John Garbuzinski
Northwestern University
Computer Science

ABSTRACT

The goal of this project is to correctly estimate the yearly fantasy football scores of NFL players given their position, past performance, and years of experience in the NFL. This is a task which hundreds of professionals attempt every year, as well as millions of fantasy football players trying to gain an edge on their opponents. With fantasy sports, and especially fantasy football, there are more opportunities than ever not only to compete with others but also to make money doing it.

1. INTRODUCTION

Fantasy football is a rapidly growing industry that in just the past few years has gone from a niche interest of hardcore sports fans to a cultural phenomenon. Millions of people sign up to play with their friends, family, and colleagues each year. While building their rosters, those people look to a select few experts, but even these experts whose full-time job is fantasy football will often make mistakes. The task of this project was to find a way to do what these experts do using machine learning. We have attempted to project fantasy football scores based on statistics from past seasons.

2. DATA

The data we used was pulled from pro-football-reference.com. We took the top thirty players in passing yards, top fifty in rushing yards, and top one hundred in receiving yards from each of the past three seasons, making a total of one hundred eighty instances per season.

2015 NFL Passing

Previous Season / Next Season

Passing ☑ Hide non-qualifiers for rate stats · Glossary · SHARE · Embed · CSV · Export · PRE · LIN

Rk		Tm	Age	Pos	G	GS	QBrec	Cmp	Att	Cmp%	Yds ▾	TD	TD%	Int	Int%	Lng	Y/A
1	Drew Brees	NOR	36	QB	15	15	7-8-0	428	627	68.3	4870	32	5.1	11	1.8	80	7.8
2	Philip Rivers	SDG	34	QB	16	16	4-12-0	437	661	66.1	4792	29	4.4	13	2.0	80	7.2
3	Tom Brady*	NWE	38	QB	16	16	12-4-0	402	624	64.4	4770	36	5.8	7	1.1	76	7.6
4	Carson Palmer*	ARI	36	QB	16	16	13-3-0	342	537	63.7	4671	35	6.5	11	2.0	68	8.7
5	Matt Ryan	ATL	30	QB	16	16	8-8-0	407	614	66.3	4591	21	3.4	16	2.6	70	7.5
6	Eli Manning*	NYG	34	QB	16	16	6-10-0	387	618	62.6	4436	35	5.7	14	2.3	87	7.2
7	Blake Bortles	JAX	23	QB	16	16	5-11-0	355	606	58.6	4428	35	5.8	18	3.0	90	7.3
8	Matthew Stafford	DET	27	QB	16	16	7-9-0	398	592	67.2	4262	32	5.4	13	2.2	57	7.2
9	Ryan Tannehill	MIA	27	QB	16	16	6-10-0	363	586	61.9	4208	24	4.1	12	2.0	54	7.2
10	Kirk Cousins	WAS	27	QB	16	16	9-7-0	379	543	69.8	4166	29	5.3	11	2.0	78	7.7

Figure 1: A selection of statistics from pro-football-reference.com

For attributes, we used the statistics commonly used in fantasy football scoring (passing/rushing/receiving yards, passing/rushing/receiving touchdowns, interceptions, and fumbles) as well as a few rate statistics to help further determine a player's quality (completion percentage, yards per carry, targets, and receptions) and each player's years of experience in the NFL to help determine whether that player's stats were due to rise or decline in the following season. We then used that data to class each player in intervals of twenty fantasy points (0-20, 20-40, 40-60, etc.) up to five hundred points, giving us a total of twenty-five classes. If a player was placed by our model into the same group that he actually finished the season in, that was considered a success.

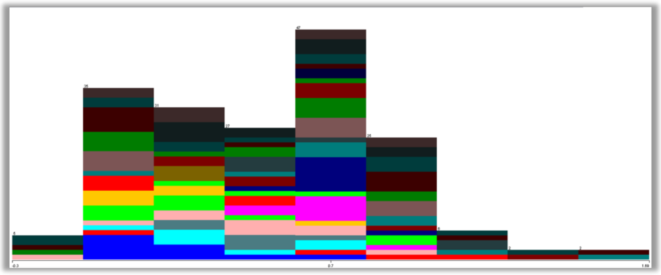

Figure 2: Visualization of testing data from WEKA, with colors representing classes

3. MODEL

To build our model, we first tested the data on various machine learning methods using WEKA 3.6. We used data from the 2013 and 2014 seasons to train the models, then tested them on data from the most recent season, 2015. The methods we tried included ZeroR (which places all instances into the most common class) to establish a baseline for accuracy, as well as decision trees, k-nearest neighbor, bayes nets, and multilayer perceptron. In their basic forms, these methods gave us respective accuracies of 6.7%, 36.1%, 62.2%, 27.2%, and 33.9%. With nearest neighbor giving by far the greatest accuracy, we decided to move forward with fine-tuning that method to see if we could improve on the results.

4. CONCLUSION

In the end, we were able to achieve an accuracy of 68.9% on our test set by adjusting the number of nearest neighbors used by the algorithm to four, and weighting these neighbors by 1/distance.

```
=== Summary ===

Correctly Classified Instances         124               68.8889 %
Incorrectly Classified Instances        56               31.1111 %
Kappa statistic                          0.673
Mean absolute error                      0.0318
Root mean squared error                  0.1492
Relative absolute error                 41.7248 %
Root relative squared error             76.3609 %
Total Number of Instances              180
```

Figure 3: Final results

Moving forward, we would suggest that anyone attempting a similar project should try to find a way to project outcomes for rookie players. Because our projections were based entirely on statistics from previous seasons, rookies (who had no stats) were not able to be projected. Additionally, some way to account for injuries (which could seriously affect a player's stats from season to season, or be more common for some players than others) could be extremely useful.