Fantasy Fumball: Using Machine Learning to Predict NFL Player Rank

Wendy Yang

February 14, 2018

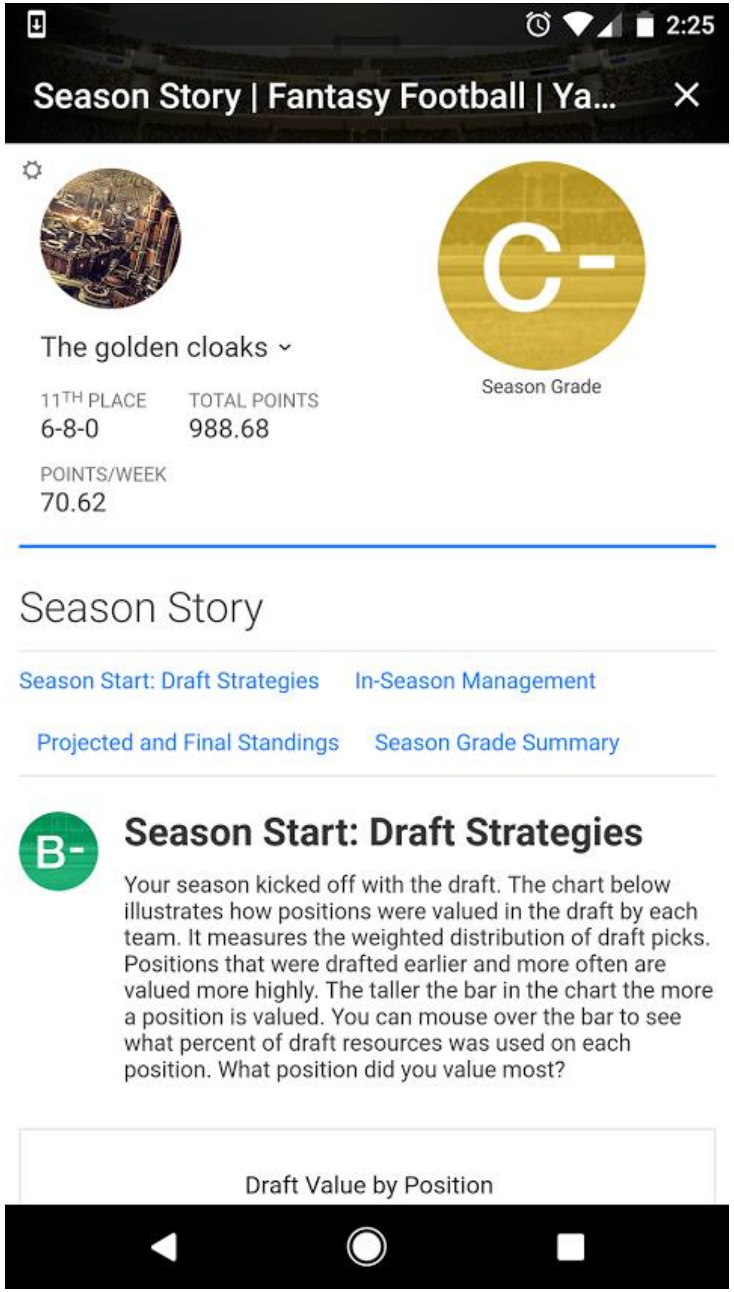
Data Science Part Time Course

Why do we care?

The Fantasy Sports industry is estimated at over $7B per year

The ability to predict player rank accurately can lead to better drafting and ultimately winning the fantasy league

Personally took 11th place in my fantasy league this year ☹



Fantasy Football Background

Fantasy Football teams are comprised of players from all teams in the NFL.

Players in the following positions may be drafted: Quarterback (QB), Running back (RB), Wide Receiver (WR), Tight End (TE), and Kicker (K).

Each player may only be drafted onto one fantasy team per league.

Each player in the league drafts one fantasy player for their team in rounds until the fantasy teams are filled.

Hypothesis

An NFL player's fantasy football rank can be predicted based on their fantasy football stats/points earned from a previous season

Why does this matter? Drafting players with the best rank should allow for the highest scoring team.

Data Collection

https://github.com/BurntSushi/nflgame

This pulls JSON files from NFL API

ESPN.com for 2016 top 300 fantasy player rank

Data exploration

This data set contained all players in the NFL, I was only interested in QB, RB, WR, TE, K

```
# Create new DataFrames using only QB, WR, RB, TE, K
nfl16_QB = nfl16.loc[nfl16.pos == 'QB']
nfl16_WR = nfl16.loc[nfl16.pos == 'WR']
nfl16_RB = nfl16.loc[nfl16.pos == 'RB']
nfl16_TE = nfl16.loc[nfl16.pos == 'TE']
nfl16_K = nfl16.loc[nfl16.pos == 'K']
```

I also ignored defensive players since we draft entire team defenses rather than individual defensive players

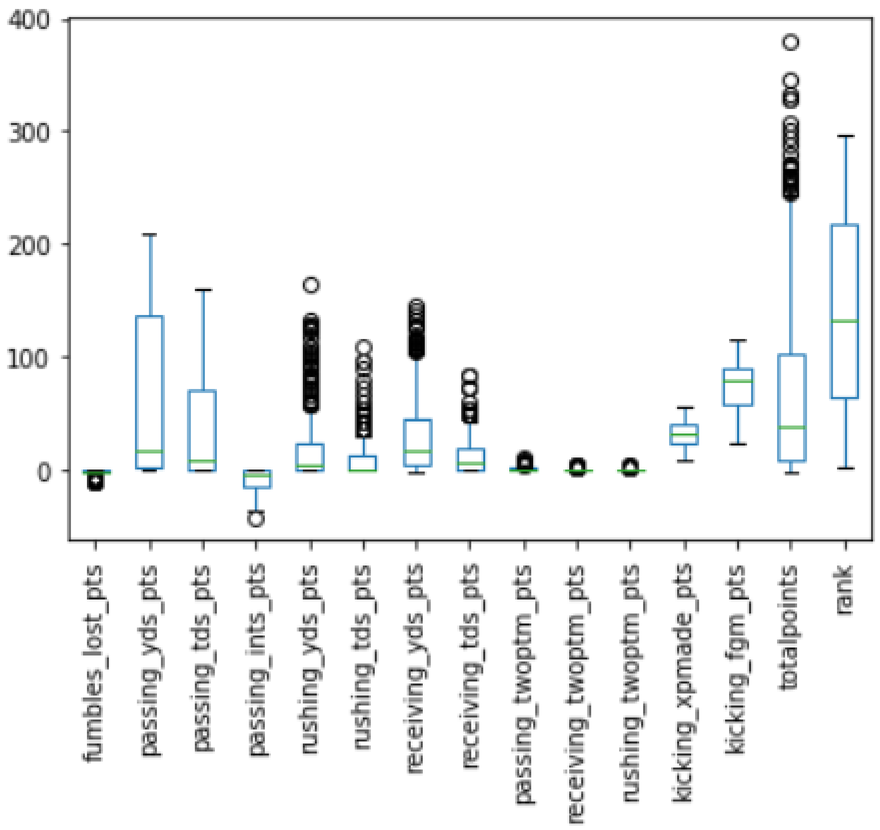
Fantasy Football Scoring

Offensive Players
• Passing Yards: 1 point per 25 yards • Passing Touchdowns: 4 points
• Passing Interceptions: -2 points
• Rushing Yards: 1 point per 10 yards
• Rushing Touchdowns: 6 points
• Receptions: 1 points (only if using PPR scoring)
• Receiving Yards: 1 point per 10 yards
• Receiving Touchdowns: 6 points
• 2-Point Conversions: 2 points
• Fumbles Lost: -2 points
• Fumble Recovered for a Touchdown: 6 points

Kicking
• Point After Touchdown (PAT) Made: 1 point
• Field Goal (FG) Made (0-49 yards): 3 points
• FG Made (50+ yards): 5 points

```python
# Map player stats to points using list comprehensions
fumbles_lost_pts = [x * -2 for x in nfl16_off.fumbles_lost]
passing_yds_pts = [x / 25 for x in nfl16_off.passing_yds]
passing_tds_pts = [x * 4 for x in nfl16_off.passing_tds]
passing_ints_pts = [x * -2 for x in nfl16_off.passing_ints]
rushing_yds_pts = [x / 10 for x in nfl16_off.rushing_yds]
rushing_tds_pts = [x * 6 for x in nfl16_off.rushing_tds]
receiving_yds_pts = [x / 10 for x in nfl16_off.receiving_yds]
receiving_tds_pts = [x * 6 for x in nfl16_off.receiving_tds]
passing_twoptm_pts = [x * 2 for x in nfl16_off.passing_twoptm]
receiving_twoptm_pts = [x * 2 for x in nfl16_off.receiving_twoptm]
rushing_twoptm_pts = [x * 2 for x in nfl16_off.rushing_twoptm]
kicking_xpmade_pts = [x * 1 for x in nfl16_off.kicking_xpmade]
# For now, counting all field goals as 3 points for ease of data
# fgyds doesn't work across entire season
kicking_fgm_pts = [x * 3 for x in nfl16_off.kicking_fgm]
```

Visualizing Data



Plotted all of the points as box plots to see if anything was an obvious predictor.

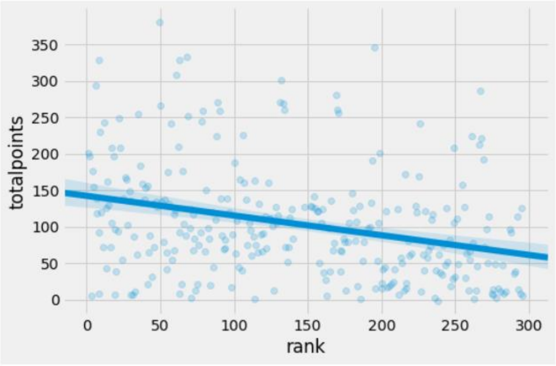
Seems like Total Points has the most outliers along with rushing/receiving yards and touchdowns

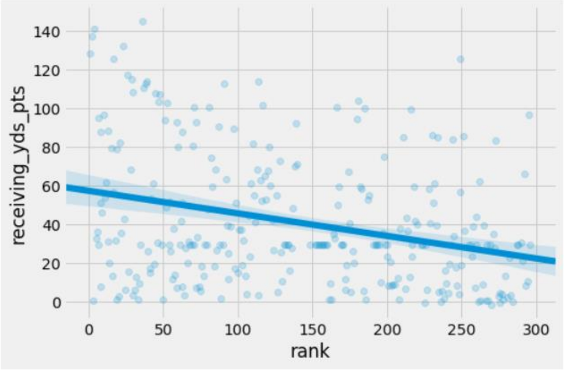
Random Forest

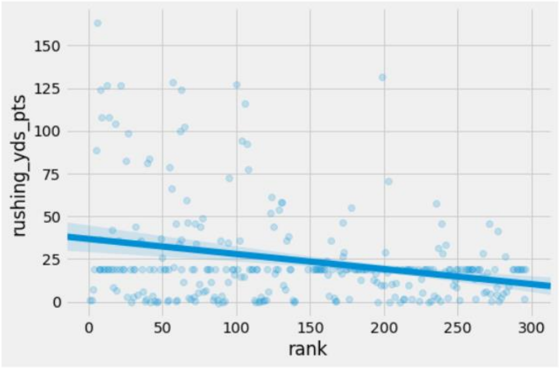
Random Forest is a quick preliminary model that can determine important features without having to do feature scaling.

Totalpoints, receiving yards, and rushing yards are the most important features in this set. All appear to be negatively correlated with rank.

	feature	importance
13	totalpoints	0.307617
6	receiving_yds_pts	0.257463
4	rushing_yds_pts	0.169146
7	receiving_tds_pts	0.086722
0	fumbles_lost_pts	0.039199
5	rushing_tds_pts	0.032189
2	passing_tds_pts	0.021820
12	kicking_fgm_pts	0.018654
11	kicking_xpmade_pts	0.018611
9	receiving_twoptm_pts	0.014018
1	passing_yds_pts	0.011532
3	passing_ints_pts	0.009027
10	rushing_twoptm_pts	0.007756
8	passing_twoptm_pts	0.006246







Linear Regression

```
print linreg.intercept_
print linreg.coef_
zip(imp_feature_cols, linreg.coef_)
```

```
211.808173066
[-0.12615973 -0.94943358 -0.81969633]

[('totalpoints', -0.12615973386161361),
 ('rushing_yds_pts', -0.94943358138773681),
 ('receiving_yds_pts', -0.81969632808597337)]
```

RMSE = 74.76

Null RMSE = 80.80

My model does better than null!

Conclusions

Random forest was able to determine the most important features

Linear regression was able to predict how player’s performance may change player rank by interpreting the coefficients.

- A player would have to gain ~1 rushing or receiving yard point to improve their rank by 1. Based on the point mapping, that means they would have to gain at least 10 more yards to improve their rank.
- A players would have to gain ~8 total points in order to improve their rank by 1.

Future Work

See if I can predict improvement in players across years.

Split apart field goals into 3 and 5 points. For now, just scored all field goals made as 3 points since I did not want to loop through all kickers of each game across the entire season to determine which kicks were longer than others.

Aggregate individual players into team defenses and pull a full team score across the season to further improve my rank predictions.

Try to draft position rank besides positions (e.g. WR1, WR2)

Turn this into a product with a user interface to select a player and determine their rank against other players. (Unfortunately, while a good lesson, "who should I pick" already exists as a product)