Fantasy Footbal Model

Jacob Cohn

12/10/2021

Introduction

Millions of people compete against their friends in fantasy sports every year. In fantasy football, a group of about twelve people will take turns selecting professional football players to add to their roster. They then use those rosters to compete against each other by seeing whose roster scores more fantasy points over the course of a week.

Objective Statement

The purpose of this project is to generate a model that will help predict fantasy point production of players given their age and their position. To get enough data to run a sufficient model, every play from every NFL game since 2001 is being included. A majority of this project is focused on cleaning and managing the data to easily run the model. This is because future projects will rely heavily on the data being created from this project. The following report will be broken into Data Cleaning, Fantasy Point Calculation, and Analysis.

Data Cleaning

Gather PBP Data

The first step in this project is to gather the records of every football play that has happened over the time period we are interested in, which can be acquired through nflfastR. This code will write every play into a csv, however the playoff games have been removed since they will bias our analysis. There are hundreds of columns that will need to be parsed but the first few rows of the description column has been displayed to show that we will be dealing with different play types, players, and yardage amounts.

```
source("src/gather_pbp.R")
head(data$desc,n=5)
```

```
## [1] "GANE"
## [2] "7-B.Gramatica kicks 59 yards from ARI 30 to WAS 11. 32-K.Watson to WAS 33 for 22 yards (44-M.Stone)."
## [3] "(14:54) 48-S.Davis right tackle to WAS 34 for 1 yard (94-M.Bell)."
## [4] "(14:15) 6-B.Matthews pass to 48-S.Davis to WAS 41 for 7 yards (98-R.Davis)."
## [5] "(13:44) 48-S.Davis right tackle to WAS 45 for 4 yards (94-M.Bell)."
```

```
rm(list=ls())
```

Gathering Rosters

Next we need to gather the roster data for all the players in the years that we are interested in. The roster will provide the data that will serve as our dependent variables such as position and age.

```
source("src/gather_roster.R")
head(rosters[,7:12],n=5)
```

full_name <chr>	first_name <chr>	last_name <chr>	birth_date <date>	height <chr>	weight <chr>
Steven Grace	Steven	Grace	1979-02-13	6-3	296
Jason Starkey	Jason	Starkey	1977-07-15	6-4	297
Nathan Hodel	Nathan	Hodel	1977-11-12	6-2	245
Mike Gruttadauria	Mike	Gruttadauria	1972-12-06	6-3	280
David Barrett	David	Barrett	1977-12-22	5-10	198

5 rows

```
rm(list=ls())
```

Aggreagting PBP Data

The aggregate_pbp.R script does most of the heavy lifting for this project. It is being run in the Rmd file but is not shown in the report because the script produces tons of checks and messages that would clutter the report. Currently the data is a list of descriptions of football plays. What the R script is doing is converting those descriptions into countable statistics that will then be summed up for each player. If we check the first few columns we will see that the number of assists for a couple of different players in a given week has been calculated but we currently don't know the players name yet.

```
summary_stats<-read.csv("data/summary_statistics.csv")
head(summary_stats[,2:5],n=5)
```

	PlayerID <chr>	week <int>	season <int>	Assist <int>
1	00-0000007	3	2002	1
2	00-0000007	7	2004	1
3	00-0000007	9	2003	2
4	00-0000007	11	2003	1
5	00-0000007	12	2003	1

5 rows

```
rm(list=ls())
```

Combining Roster Data and Summary Statistics

To get the names of the players whose stats are given in the table above, we need to merge the stats with the roster by PlayerID. This code looks up the PlayerID's and adds the appropriate name, team, height, weight, etc. It also calculates each player's age based on their birthday and what the year was in the season they were playing.

```
source("src/combine_stats_and_roster.R")
head(filled_stats[,c(1,2,3,4,5,9,63)],n=5)
```

	PlayerID <chr>	season <int>	position <chr>	team <chr>	full_name <chr>	week <int>	Tackle <dbl>
1	00-0000007	2002	RB	CHI	Rabih Abdullah	13	1
2	00-0000007	2002	RB	CHI	Rabih Abdullah	10	2
3	00-0000007	2002	RB	CHI	Rabih Abdullah	3	1
4	00-0000007	2002	RB	CHI	Rabih Abdullah	11	1
5	00-0000007	2002	RB	CHI	Rabih Abdullah	5	NA

5 rows

```
rm(list=ls())
```

Calculating Fantasy Points

Now that we have the data in a neat and understandable format, it's time to calculate how many fantasy points each player scored. This section of code is dependent on two files. The first is the weekly_fantasy_points.csv file that we created in the data folder during the last function. The second is the points.csv which says how many fantasy points each play is worth. This function will multiply each players stats by how much that stat is worth. All those values are then summed up into the FPoints column to denote how many Fantasy Points the player scored that week. Below is a subset of the data to show what Fantasy Points production looks like in a given week.

```
source("src/calculate_Fantasy_Points.R")
head(weekly_sums_df[,c(2,3,4,5,6,10,66)],n=5)
```

	PlayerID <chr>	season <int>	position <chr>	team <chr>	full_name <chr>	week <int>	FPoints <dbl>
1	00-0000007	2002	RB	CHI	Rabih Abdullah	13	1.75
2	00-0000007	2002	RB	CHI	Rabih Abdullah	10	1.00
3	00-0000007	2002	RB	CHI	Rabih Abdullah	3	1.57
4	00-0000007	2002	RB	CHI	Rabih Abdullah	11	0.50
5	00-0000007	2002	RB	CHI	Rabih Abdullah	5	-2.00

5 rows

```
rm(list=ls())
```

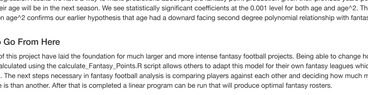
Analysis

We finish by testing to see what effects age and position you play have on your fantasy point production. So, we will load in the weekly fantasy point statistics that we generated in the last function and sum up their production for the full year.

```
season_df <- fantasy_df%>%
  dplyr::group_by(PlayerID,season, position,team,full_name,age)%>%
   dplyr::summarise_at(sum_cols, sum, na.rm=TRUE) %>%
  dplyr::ungroup()
```

We are printing out a scatter plot of every player in the dataset with the X axis showing how old they are and the Y axis showing how many fantasy points they scored. Each point is a unique players fantasy point production in a single year. The darker shade shows a higher density of players. When we look at the graph we see a trend of fantasy points increasing with age until they start reaching their physical peak at 27 years old and then start to decrease.

```
plot(season_df$age,season_df$FPoints,xlab = 'Players Age',ylab = 'Fantasy Points',xlim = c(20,50),col=rgb(.6,0,0,0.065),pch=16)
```



We are going to create and age^2 value for each player since there seems to a be a parabolic relationship between age and fantasy points.

```
season_df$age2 <- season_df$age^2
```

Here we are running a fixed effects model on the data to see what the coefficients of age and age^2 are. We are using a fixed effects model because we want each player to have their own intercept. This allows us to make predictions about individual players point production. We include age^2 because the graph of the relationship between age and fantasy points shows a downward facing parabola.

```
fixed.dum <- lm(formula = FPoints ~ age + age2 +factor(position) + factor(PlayerID),data = season_df)
```

```
summary(fixed.dum)
```

```
##
## Call:
## lm(formula = FPoints ~ age + age2 + factor(position) + factor(PlayerID),
##     data = season_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -409.05   -7.10    0.00    6.72  307.40
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -136.45741   26.09238  -5.230 1.71e-07 ***
## age                             13.07644    0.88553  14.767  < 2e-16 ***
## age2                            -0.26811    0.01578 -16.991  < 2e-16 ***
## factor(position)CB             -13.24599   10.80915  -1.225 0.220419
## factor(position)DB             -21.83158   10.70213  -2.040 0.041367 *
## factor(position)DE              -4.10133    9.18591  -0.446 0.655254
## factor(position)DL             -18.57153   28.83892  -0.644 0.519597
## factor(position)DT              -5.23571    9.28361  -0.564 0.572777
## factor(position)FB              6.72215    9.53854   0.705 0.480981
## factor(position)FS              -8.17493   10.82763  -0.755 0.450252
## factor(position)G              -2.47247    3.06544  -0.807 0.419924
## factor(position)HB             18.48449   20.57726   0.898 0.369035
## factor(position)ILB             8.79880    9.37486   0.939 0.347969
## factor(position)K              74.54290   42.54683   1.752 0.079781 .
## factor(position)LB             -7.82452    9.12502  -0.857 0.391188
## factor(position)LS             13.14408    7.17937   1.831 0.067146 .
## factor(position)MLB            10.62942    9.45096   1.125 0.260729
## factor(position)NT             -2.26566    9.58223  -0.236 0.813090
## factor(position)OG             -0.31844    3.60207  -0.088 0.929555
## factor(position)OL             3.36076    5.84670   0.575 0.565422
## factor(position)OLB            3.74375    9.18664   0.408 0.683628
## factor(position)OT             -1.61809    4.17182  -0.388 0.698121
## factor(position)P              69.56059   56.91503   1.222 0.221649
## factor(position)QB            -50.76355   19.98309  -2.540 0.011080 *
## factor(position)RB             0.06883    9.49859   0.007 0.994337
## factor(position)SAF           -24.25467   12.32658  -1.968 0.049116 *
## factor(position)SS             -6.52314   10.78265  -0.605 0.545206
## factor(position)T              -3.42819    3.81263  -0.899 0.368571
## factor(position)TE             1.58485    8.54824   0.185 0.852917
## factor(position)WR            -21.49391   10.91738  -1.969 0.048988 *
## factor(PlayerID)00-0000017     35.62482   30.54786   1.166 0.243545
## factor(PlayerID)00-0000032    -11.30776   34.12636  -0.331 0.740383
## factor(PlayerID)00-0000045     -6.97801   25.87444  -0.270 0.787403
## factor(PlayerID)00-0000065     13.03644   26.52670   0.491 0.623110
## factor(PlayerID)00-0000100     21.66602   30.26159   0.716 0.474023
## factor(PlayerID)00-0000108     46.00215   37.34776   1.232 0.218063
## factor(PlayerID)00-0000112      7.50588   33.84057   0.222 0.824470
## factor(PlayerID)00-0000121      1.50069   26.13033   0.057 0.954202
## factor(PlayerID)00-0000210 128.37977   41.41810   3.100 0.001940 **
## factor(PlayerID)00-0000217      4.48484   27.96965   0.160 0.872609
## factor(PlayerID)00-0000231      4.18606   42.29146   0.099 0.921154
## factor(PlayerID)00-0000242 -19.77642   41.97371  -0.471 0.637529
## factor(PlayerID)00-0000251  75.94334   26.45788   2.870 0.004103 **
## factor(PlayerID)00-0000261  51.22569   30.55505   1.677 0.093651 .
## factor(PlayerID)00-0000282 109.24190   39.52448   2.764 0.005715 **
## factor(PlayerID)00-0000313 102.46507   41.41446   2.474 0.013411 *
## factor(PlayerID)00-0000339 104.34159   29.52378   3.534 0.000410 ***
## factor(PlayerID)00-0000352 -10.13403   26.65039  -0.380 0.703757
## factor(PlayerID)00-0000374 -14.93483   27.95945  -0.534 0.593222
## factor(PlayerID)00-0000400 -62.49434   57.96524  -1.078 0.280983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.87 on 27572 degrees of freedom
##   (1095 observations deleted due to missingness)
## Multiple R-squared:  0.7588, Adjusted R-squared:  0.6815
## F-statistic: 9.825 on 8827 and 27572 DF,  p-value: < 2.2e-16
```

With these regression results we have a way to make predictions about players fantasy point production given their previous years performance and what their age will be in the next season. We see statistically significant coefficients at the 0.001 level for both age and age^2. The negative coefficient on age^2 confirms our earlier hypothesis that age had a downard facing second degree polynomial relationship with fantasy point production.

Where To Go From Here

The results of this project have laid the foundation for much larger and more intense fantasy football projects. Being able to change how fantasy points are calculated using the calculate_Fantasy_Points.R script allows others to adapt this model for their own fantasy leagues which use their own rule set. The next steps necessary in fantasy football analysis is comparing players against each other and deciding how much more valuable one is than another. After that is completed a linear program can be run that will produce optimal fantasy rosters.