Final Report:

NFL Fantasy Football Player Predictions

Sports betting is one of the fastest growing markets both in the US and globally. Sports betting is pretty much what it sounds like, betting that you can predict the future when it comes to the sports or athlete of your choice. Some people bet on the scores of a soccer game, which horse will win the Kentucky Derby, how many strikes a pitcher will have in a baseball game, or who will win the NBA championship. Some places even take bets on how long the national anthem will last for in any given game, or how many outfit changes a halftime performer will make at the super bowl! Believed by some to be a $150+ Billion industry (including illegally), legal sports betting was valued at nearly $70 Billion USD in 2020, with an annual growth rate expected to be over 10% a year, hitting over $140 Billion by 2028. In 2019, the state of Nevada in the US reported over $5.34 billion in that state alone.

Up until 2018, the only state in the US you can legally bet or gamble in was Nevada. In 2018, the supreme court ruled in favor of allowing other states to participate in the legal sports betting. Since then, Americans in the dozens of newly eligible states have wagered over $65 billion in sports betting, with that rate growing increasingly fast. With online sports betting being as easy as just an app download and a few clicks, anyone can now join in on the fun!

Goldman and sachs predict online sports betting will grow at a massive 40% growth rate annually over the next decade. I believe that because of all of these factors, there is massive potential to capitalize on this new growing market. More specifically, I am interested in the sport that owns the largest share of the sports betting pie: Football.

I believe that the need for accurate sports predictions for sports is at an all time high. The most popular and highest demand sport I have attempted to predict is Football, more specifically Fantasy Football. I developed 4 different models, for the 4 different positions I was interested in for football, that was able to predict a football player's end of season total fantasy points on average within 27.5- 67.8 points, depending on position. Over a 16 game season, this comes out to about 1.71 - 4.23 points off per week.

Data

The goal of my model was to use players' previous data points to predict a future season. I started the analysis with a very large data set consisting of not only weekly player data from 2000-2019, such as catches and yards in a given week in a given season, but also physical player data such as height, weight, combine 40 yard dash time, bench press, etc. For my models, I was not interested in player physical traits, so I focused primarily on players' weekly and yearly statistics, which ranged all the way back to 2000. One limitation I had on the data was that the player “snap count” rate was not tracked until 2012, so I limited my dataframe to contain only the player statistics from 2012 onwards. To condense my data frame a bit further, I grouped players by player ID, name, team, and season. I then used the numpy sum function to get the total statistics for a player for a given season, as well as the numpy mean function to get the average snap count percentage and average QB passer rating for a season. These variables would help me with the next step in my exploratory data analysis.

The next step I performed was creating a for loop for my data so I could find the lag features in my data. For example, lag_receiving_tds_1 would be touchdowns from 1 year prior to 2019, lag_passing_tds_3 would be passing touchdowns from 3 years ago. This was useful for me to find out which past variables have the highest correlation to the variable I was looking to predict: Half PPR fantasy points.

By this step I realized it was important to look at positions separately. What is important for a Quarterback scoring fantasy points, is very different from what is important or relevant to a wide receiver. I separated my dataframe into 4 separate dataframes depending on position, while also limiting my dataframes to a minimum of 50 snaps from the previous season (lag_offensive_snapcount_1). This I believed was the best thing to do, so players who did not perform enough to impact their team or played minimal snaps were removed from impacting the training of my models.

One limitation to this method is that incoming rookies with no previous data points, or players who played minimal snaps or who might have gotten injured, are not accounted for in this model or predicted. Because of the variance that comes with attempting these players from a lack of previous data, I believed it would be worth the trade off to remove these players to better predict the players with more previous data points.

Finally, I ran a correlation matrix for my 4 dataframes to find out which lag features had the greatest correlation to fantasy points for each specific position. The features with the highest correlations are an indication of which features I will use in the next step for my modeling portion.

(Correlation Matrices for each position)

Modeling

After finding the features with the highest correlation, I began my modeling. The strategy I used was the same process for the 4 separate positions:

1. Established an X (list variables I am used to predict, using the features from the correlation matrices) and a y (always looking for half_ppr_fantasy_points)
2. Ran a train_test_split to get an X_train, X_test, y_train, and y_test for each position separately
3. Trained a Dummy Regressor Model with a strategy of "mean" to get a baseline for my other Regression models
4. Trained and fit a Linear Regression Model, then used it to predict the position
5. Fit a Grid Search CV model to find the optimal number of trees to use in a Random Forest Regression model
6. Finally, I trained and fit a Random Forest model (while using the optimal n_estimators from the grid search, and used it to again predict a position

For each position, I trained and fit 3 separate models on it ("Mean" Dummy Regressor, Linear Regression, and Random Forest) and compared the Mean absolute error for each position. For every position, both the Linear Regression Model and Random Forest Model outperformed the dummy regressor model by a large amount. However, for each of the 4 positions, the Linear Regression model outperformed the Random Forest model and had the lowest mean absolute error across each position.

(**Comparing Mean absolute error for each model, for each position)**

(top 24 predicted players based on Linear Regression model, Random Forest Model) ((actual 2020 data as well?)

Some take aways:

-The Linear Regression model outperforms the Random Forest model for each position, but only slightly. Both of these models out perform the 'mean' dummy regressor model by a good amount

-As stated before, the model has a hard time accounting for rookies, younger players, or anyone with few or no previous data points.

-Model often under- predicts the top players, and over predicts the players towards the bottom of the list near rank 100 per their respective position. The top players each season are often outliers that outperform what is expected of them, so all of these models seem to not account for that when projecting players.

-Model does not factor in players specific ability. Some players are just more talented and out-perform these predictive models

How to apply this to Fantasy Football

The best way to apply this in my opinion, would be to first divide the players total projected Fantasy Points by 16 (number of weeks in a season), so you can get an average for each game. Fantasy football is played weekly, so this is just a starting point. There are theoretically 1000's of variables that impact a player formance on any given day. For further modeling or future research, I think the next step I would take would be to customize this weekly projection based on the strength of opponents played. if an opponent gives up the least amount of yards on average over a season for example, the player would be projected less for that specific weekly game.