Fantasy Football Modeling (Dream Team)

Darren Chang, Jack Vaughan, Zach Schaffer

1 Introduction

Fantasy football leagues are composed of teams that compete head-to-head with lineups composed of (depending on league settings) a combination of quarterbacks (QB), wide receivers (WR), running backs (RB), tight ends (TE), a defense, kickers, or flex positions filled by receivers, running backs, and tight ends. For each week of the NFL regular season, each NFL player is given a projected point score to help fantasy football players select the best lineup possible for that week. The goal of this project is to find a method to accurately predict fantasy football points each week. These predictions would be helpful for fantasy football connoisseurs as well as bettors who are placing bets and making markets on fantasy football games or the performance of different players.

While perhaps this problem appears simple, note that predictions cannot be generated for a game in week k with data from week k; rather, the player statistics from weeks 1 to $k-1$ are used, as only that accurately reflects the available information. Our approach to creating features useful for forecasting is described in section 2.2. We develop a variety of linear models for generating predictions and a low rank model missing data imputation for this purpose and discuss different measures of error for each model.

In addition, we "backtest" by comparing our predictions to ESPN predictions for the 2019 NFL season. We find that measures for cumulative and previous week player statistics and proxies for team strength are reasonable predictors of fantasy football points. While our method is not strictly better than other predictions such as those by ESPN, we provide a starting point for an open-source method to accurately predict fantasy football points with Machine Learning.

2 Data

2.1 Dataset Description

We used three primary datasets used for this project. The first is weekly fantasy football data from 1999 to 2019 sourced from Fantasy Football Data Pros, which contains fantasy football points and game statistics to compute those points. About 200-300 players are represented in the data for the 17 weeks of the regular NFL season. The three measures of fantasy football points for each player include the standard calculation (one point for ten yards of offense generated with bonuses for touchdowns and deductions for interceptions and fumbles), the points per reception (PPR) calculation (the standard rules with the modification that each reception is an extra point), and the half PPR calculation (the PPR rules except that each reception is only a half point). Offensive statistics such as passing yards, rushing yards, and reception yards, completions, interceptions, and touchdowns are included in the data. The overall quality of the data is high, as no player is missing information; however, players vary from week to week and there is little information at different levels of granularity. In addition, the first week of the season has little information available. In the base version of the data, player statistics are inputted as 0 values; missing data imputation is discussed later in this report. Approximately 6% of the data is from the first week.

The second dataset is game-level data from 2009 to 2019 from the `nflfastR` package, which is available from CRAN. This dataset contains a `game_id` field, which can be linked to other game-level stats, scores, and the teams involved in each game.

The third dataset is game-level elo data from FiveThirtyEight, which was added as a proxy for team strength. Teams gain rating when they win against a team with a higher rating and vice versa. In contrast to one-hot encoding for each of the 32 NFL teams, this method reduces multicollinearity. The data contains an elo rating for each team before the game is played, which fits with the forecasting nature of the project.

2.2 Cleaning

These three datasets are joined together so that each player entry for each week has player statistics, game data, and elo data. Players without position data or with non-standard positions are dropped from the data, and players given hybrid positions (e.g. WR/PR) are sorted into one of the seven offensive positions that score fantasy points. Team moves, such as the Chargers' move from San Diego to Los Angeles in 2017 are accounted by creating a matching dictionary for the NFL teams across all seasons. After cleaning, there are $n = 51,622$ player-weeks

represented in the data, with nominal, discrete, and continuous values for features.

2.3 Feature Engineering

For fantasy football projections, feature engineering is a critical part of transforming the data to be used in models because projections must be created based on data from the previous weeks. Two sets of features are engineered for each player for projection purposes:

1. **One week lag** data, which are game statistics from the previous week and are denoted with the suffix `_prev`. For the feature set $X_{i,t}$, $X_{i,t,prev} = X_{i,t-1}$.
2. **Season cumulative** data, which are game statistics averaged over the season and are denoted with the suffix `_cum`. For the feature set $X_{i,t}$, $X_{i,t,cum} = \frac{\sum X_{i,t}}{t}$.

We think that these are useful features for predicting player fantasy performance in a specific week because the lag represents a "momentum" factor for how well a player is performing at a specific point in the season and the cumulative data represents an additive "baseline" factor for how well a player is performing throughout the season. There are 32 features for player and team statistics and one-hot encoded columns for player position.

2.4 Distribution of Data

The histogram in Figure 1 displays the distribution of yardage for passers, rushers, and receivers for both the one week lag data and season cumulative data. The passing data appears approximately normally distributed, as there is a wide spread for the number of passing yards quarterbacks can pass for in a week. However, the data for rushing and receiving yards are right-skewed (long right tails) because many rushers and receivers are concentrated towards the lower end of the distribution for yardage. This could be because some observations are from third or fourth string running backs or receivers who are brought in for just a few rushing attempts.

Figure 2 displays the Pearson's correlation plot of the base features of the data. This plot does not show feature engineered data, which is highly correlated with raw data of the same base type. For example, `RushingYds` is highly correlated with `RushingYds_prev` and `RushingYds_cum`. Data sourced from the same position are also highly correlated. For a quarterback, for example, `Cmp` (completions), `PassingYds`, `PassingTD` (passing touchdowns), and `Int` (interceptions) are all highly correlated, as shown by the dark blue shading. However, features between positions are not correlated: quarterback features such as completions are not nearly uncorrelated with receiving features like `Tgt` (targets).

Finally, figure 3 shows the density of the three different fantasy point measures that compose the output space. Each measure appears right skewed, with a peak between 0 and 5 points scored. Negative values also exist for each point measure. The standard point calculation has the highest density at the lowest point values; in fact, the PPR and half PPR calculations were developed to correct to smooth the distribution of fantasy point values.

Figure 1: Histograms of Previous and Cumulative Yardage

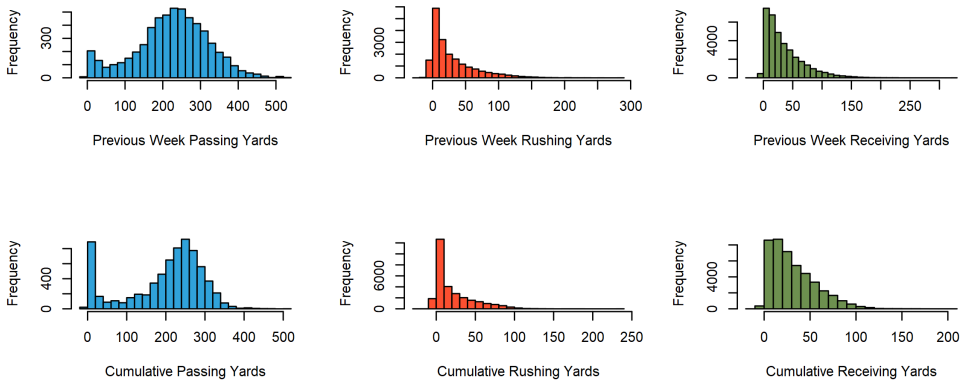


Figure 2: Correlation of Features (Truncated)

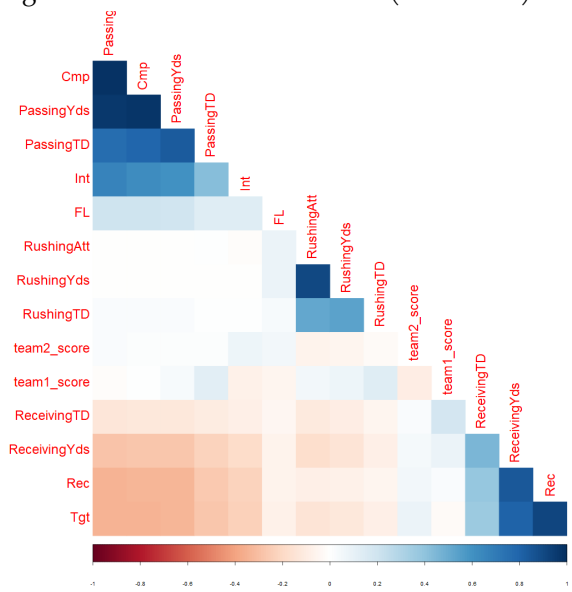
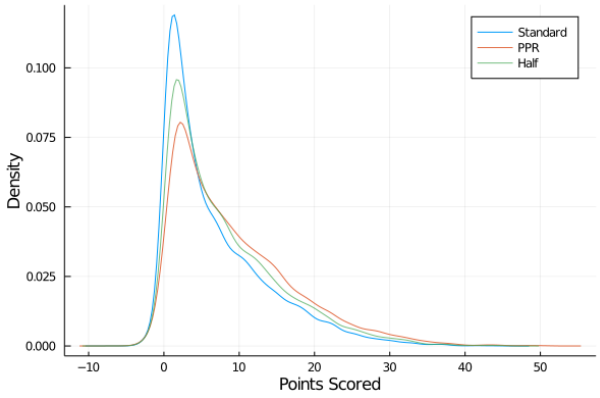


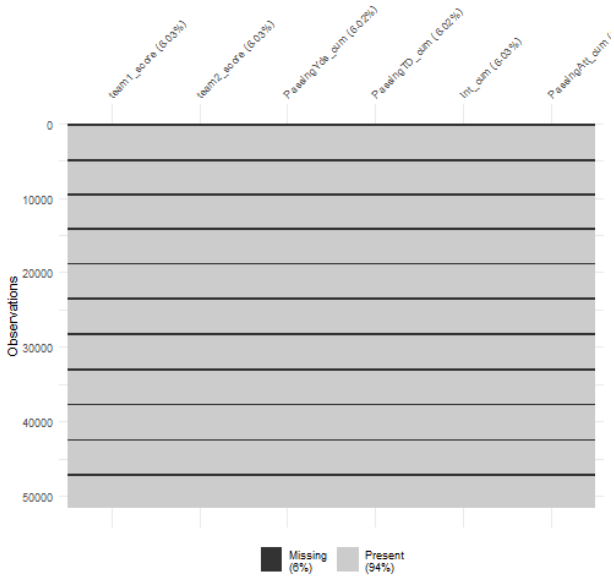
Figure 3: Density of Distribution of Fantasy Points



2.5 Missing Data Imputation

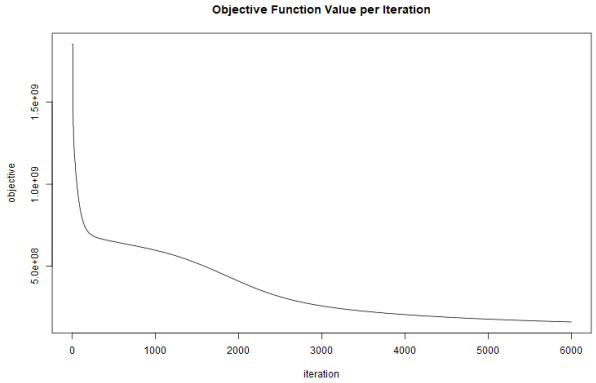
As described above, week 1 of any season lacks data (see figure 4); in fact, the only available information from the previous season is the elo ratings, which are kept for the entirety of a franchise's history. While we can think of every season as a discrete prediction space, players and teams are (somewhat) consistent from season to season, meaning that discretizing this problem may not accurately reflect its composition. In our base case, the `_cum` and `_prev` features are given 0 values and in the imputed case, the features are giving missing values.

Figure 4: Missing data Data (Selected Features)



We use a low rank model to impute the missing values. First, we used the scree plot criterion from a Principal Components Analysis procedure to decide on a rank of $k = 5$ to use in the Generalized Low Rank Model (GLRM). Second, we fit a GLRM model using the `H2o` package in R to our data set with missing values, which converged at approximately iteration 4000 (see figure 5).

Figure 5: Objective Function value and Iteration



We use a quadratic loss function for each feature and nonnegative regularization, as the predicted player statistic are nonnegative. After all, very few players in the NFL are so incapable that they manage -5 yards in a game. Third, we used the results of the GLRM model to impute missing data values (and only missing data values) to our original feature matrix.

The results are mostly reasonable but somewhat uninformative, at least partly due to the lack for data

available for week one in the first place. Moreover, the imputed data creates some incorrect or unintuitive results; for example, quarterbacks were often predicted to have nonzero cumulative receptions. Using the imputed dataset, we ran the supervised models described in the analysis section. While results of are discussed in this final report, the error metrics are nearly identical to the error metrics with models trained on the data where missing values were coded as 0.[1]

Imputing missing values with the current method is only slightly net better for prediction. This has three implications. First, perhaps discretizing seasons and placing 0 values for week 1 is the correct interpretation for this problem — in effect, this would claim that players "start over" with each new season and that finding representative player statistics is incorrect. Second, different low rank models with different settings or missing value imputation methods could be explored to better address the missing data. Third, other game-level features beyond the elo rating for each team could be added such that the data is less dependent on individual player statistics. However, this may dilute the predictive power of the data if too many features are added.

3 Analysis

For analysis, we split the data into a training set with 80% of the data and a test set with the remainder. We pursued three models using each of the three points calculations as the output space: (1) the ordinary least squares l_2 regression, (2) lasso regression and (3) quantile regression using the `GLM.jl`, `Lasso.jl`, and Scikit learn library in both Python and Julia.

3.1 OLS Regression

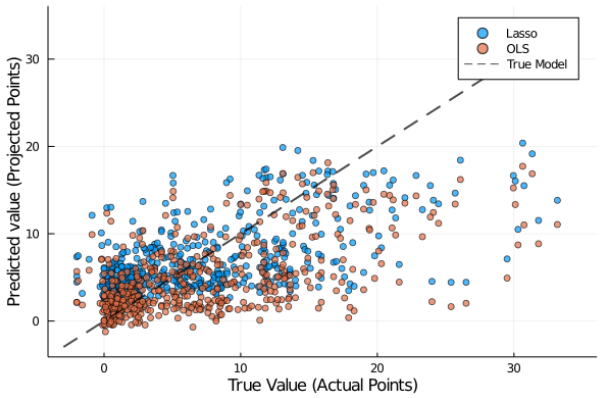
First, we construct a baseline linear model using quadratic loss without any regularization functions. The objective function for this model is

$$\min \sum_{i=1}^{n} (y_i - w^T x_i)^2 \tag{1}$$

A previous version of this analysis found that using one-hot encoding to compute the 64 features created data that was not linearly dependent; as a result, the calculated coefficients were not unique and the low-rank setting needed to be turned on to compute any coefficients at all. However, eliminating the multicollinear coefficients and representing those features in a different way would be more faithful to the method and original data. Thus, the data used for this procedure did not contain one-hot encoding for each of the 32 NFL teams and instead used the elo rating as a proxy for team success.

We use 10-fold cross-validation and take the mean of the coefficients to minimize the effect of overfitting. Figures 6 through 8 display the scatter plot of predicted values with the OLS and lasso regression on the test dataset compared to the reported fantasy points scored. The test mean squared errors (MSE) for the different output spaces are $MSE_{std} = 37.84$, $MSE_{ppr} = 52.65$, $MSE_{half} = 44.41$ and the R^2 values are between 0.16 and 0.20. With the missing-imputed data, the MSE values are $35.45, 49.01,$ and 41.43 respectively with R^2 values between 0.21 and 0.24. For each player-week, that means that the error for the standard model is about 6 points. This is not insignificant for lower-tier players who may score between 5 and 10 points per game, but is reasonable for higher-tier players — think of superstars such as Kansas City Chiefs quarterback Patrick Mahomes, Seattle Seahawks wide receiver DK Metcalf, or Tennessee Titans running back Derrick Henry — who regularly score 20 or more points per game. From these results, we concluded that the basic linear model was a first-pass attempt but further tuning and regularization may be needed for accurate point predictions.

Figure 6: Predicted versus Actual Values for Linear Regression (Standard Points)



[1]For more details on models trained with the imputed data, refer to the notebook on our Git repository: https://github.com/jck249/dream-team/blob/main/base%20models/lm_base_imputed.ipynb.

Figure 7: Predicted versus Actual Values for Linear Regression (PPR Points)

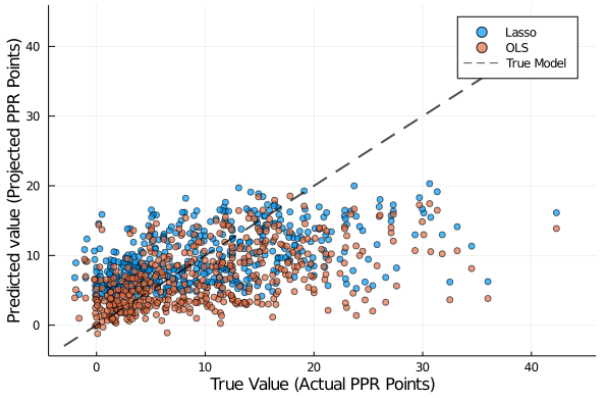
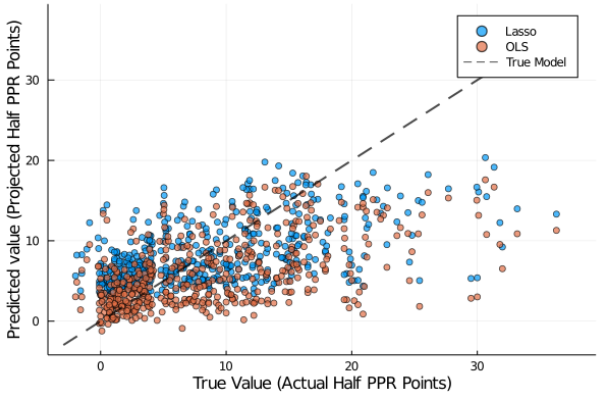


Figure 8: Predicted versus Actual Values for Linear Regression (Half PPR Points)



3.2 Lasso Regression

Next, we fit a lasso model onto the training data and use 10-fold cross-validation to find the λ value that minimizes the MSE. The objective function for this model is

$$\min \sum_{i=1}^{n} \left(y_i - w^T x_i\right)^2 + \lambda \sum_{i=1}^{n} |w_i| \tag{2}$$

Lasso regression implements shrinkage, which encourages simple and sparse models and helps reduce overfitting by fitting a model only on a subset of features. The cross-validation procedure for the basic dataset selected a lasso model with only 12 of the 37 coefficients, which corresponds with $\lambda \approx 0.35$. These 12 coefficients were all selected across each of the output spaces, which lends credence to the idea that these 12 features contribute the most to predicting fantasy football points, no matter the metric used to calculate points. The list of coefficients include the cumulative and previous week yardage for passing, rushing, and receiving, previous and cumulative team scores and elo scores. The MSEs for the standard data, PPR data, and Half PPR data are $33.71, 46.43$, and 39.22, respectively.

For the imputed dataset, the lasso regression only selected 9 coefficients — the 12 coefficients for the regression with the base dataset but without the cumulative team score and opposing team scores and the opposing team elo score. The MSEs for the standard data, PPR data, and Half PPR data are $32.43, 44.59$, and 37.67, respectively. The lower test MSE metric for both lasso models compared to the MSE for the OLS regression indicate that the lasso model may be a better fit for prediction, possibly because the lasso regression eliminates features to reduce overfitting and noise. Figures 5-7 (left) overlay the OLS regression on the lasso regression and show that lasso regression is less sensitive to outliers compared to the l_2 regression, as demonstrated by the shorter distance from underpredictions (on the right) for the lasso model to the true model line.

3.3 Quantile Regression

In order to expand the analysis of the two previous linear models, we used quantile regression on the base data set. The objective for quantile regression is given by

$$\min \frac{1}{n} \sum_{i=1}^{n} \alpha \left(y_i - w^T x_i\right)_+ + (1-\alpha) \left(y_i - w^T x_i\right)_- \tag{3}$$

where α is the value of the quantile.

These quantiles give insight on how the regression is affected by different quantiles. We regress the training features on the output space using quantiles between $q = 0.05$ and $q = 0.95$ and compare these results to the OLS model. Figure 9 shows the given quantiles and the results of the quantile regression per quantile and the associated OLS model below. These quantile regression estimates are graphically displayed; each dotted line represents the regression for a given quantile. Finally, Figure 10 contains two functions; the black lines are 10 quantile regression estimates, the dotted lines represent 95 percent confidence intervals, and then the red line represents the OLS regression results pairing with its correlating 95 percent confidence interval. The overlapping characteristic of Figure 10 allows for analysis on how different quantiles affect the overall regression compared to other quantiles of the complete data.

Figure 9: Predicted versus Actual Values for Quantile Regression (Standard Points)

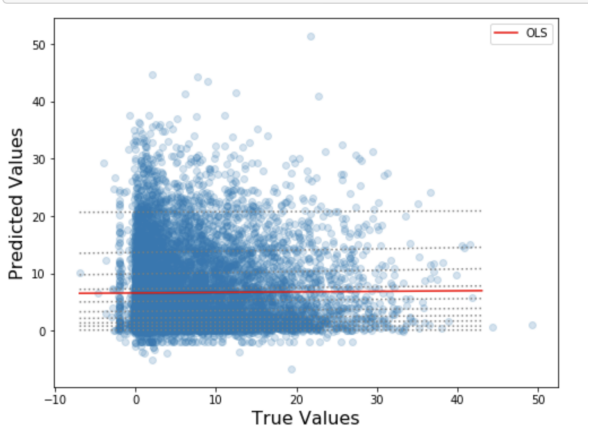
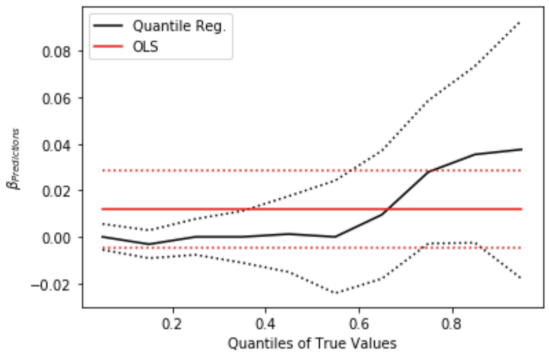


Figure 10: Quantile versus OLS Regression



To interpret the comparison of the predicted standard points and the actual points received per player given in Figure 9, we first recognize the close density of the bounds when $q \leq 0.55$. This indicates that the data below the OLS graph is much more compact and closer in distance to the regression line compared to data above it. This is likely due to the higher likelihood for a randomly selected player to underperform than overperform. The players in these lower quantiles have very close ranges compared to the upper quantiles; the difference in bound between $q = 0.15$ and $q = 0.25$ is 0.584 points, while the difference between $q = 0.75$ and $q = 0.85$ is 3.746 points. This is due to the right-skewed nature of the histogram of scored points and the fact that a large majority of players of the NFL score just shy of the OLS prediction. What drags this model up is the few star players of the league that preform exceptionally well. Since the number of players that perform below the OLS predicted value is larger compared to the number of players that perform above the OLS, the graph moves closer to the lower quantiles, and the given bounds are much more dense due to the higher quantity of players in this range.

Figure 10 gives insight onto how each quantile of fantasy points affects the OLS. As stated earlier, the dotted lines indicate a 95 percent confidence interval (CI). When the 10 quantile loss estimate function is outside of the OLS confidence interval, the range of quantiles outside the CI have a statistically significant effect whereas the function inside of the confidence interval does not have a significant impact.

Our comparison graph indicates that for lower quantiles, the OLS is not significantly altered by their values. The solid quantile estimates come very close to breaking the lower confidence bound at $q = 0.1$; however, these values are not statistically significant. The quantile regression estimator breaks the OLS confidence interval for quantiles greater than or equal to $q = 0.72$. We think this is due to the super star effect where the players in this range earn such higher points compared to the margin of the $(1 - q)$ quantiles. the OLS is then drastically affected by these players. The line almost crossing at $q = 0.1$ is likely due to the great quantity of players that exist in this region and the slight margin each player contributes to dragging the model down.

3.4 Backtesting

To compare our projections against current projections, we test our predictions against ESPN's projections for the 2019 season[2] sourced from Fantasy Football Data Pros using the imputed dataset. ESPN's projections use the standard point computation, and after joining our projections to the ESPN dataset by week and player name, there are $n = 1,938$ player weeks in the data. We find that while our models perform worse on average, they show promise in specific weeks and are less skewed than ESPN's models.

Figures 11 and 12 display histograms of the difference between the true fantasy football points and different projections. Interestingly ESPN's models appear positively biased; on the other hand, both of our models have a median (based on the histogram) closer to 0. However, the downfall of our model is in the skewness: we are more likely to underpredict and with higher magnitude.

[2] 2019 is the only season for which data is available.

Figure 11: ESPN versus OLS

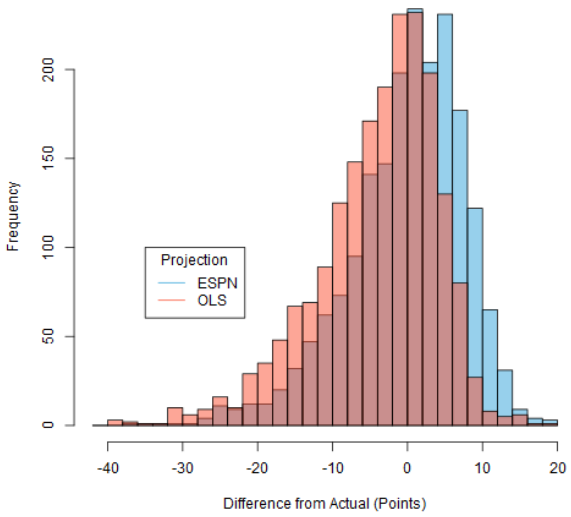


Figure 12: ESPN versus Lasso

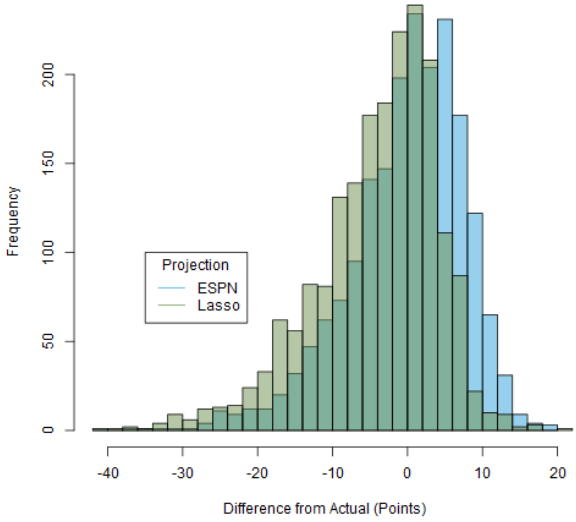
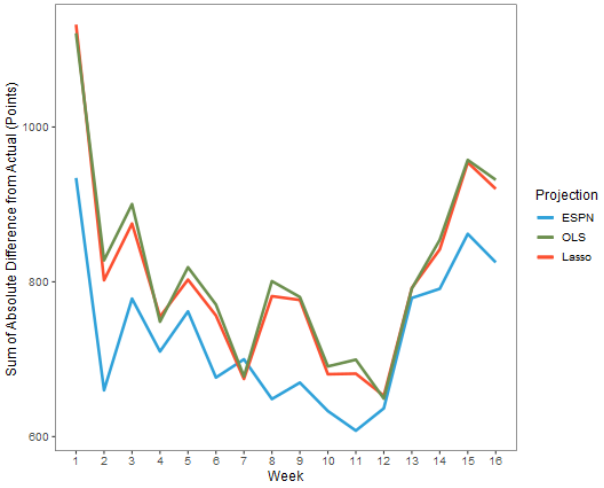


Figure 13 displays a weekly comparison between the mean absolute error, which lends credence to the idea that while our projections are not better on net, the error of the models are not far apart, especially in later weeks. While ESPN's methods are black-box, their better success in week 1 (although week 1 is difficult to predict for all models) is likely due to human imputation of week 1 data based on analyst knowledge (or on previous season or college data).

Figure 13: Absolute Error Comparison by Week



4 Conclusion

In this project, we have developed a method to predict fantasy football points for a specific player with a set of features for each game. We use low rank models for missing data imputation (for week 1 data), a plethora of linear models for predictive purposes, and methods such as test/training set splits and cross-validation to reduce the effect of overfitting. In addition, we compare our model results to those used by ESPN and find that while our models do not outperform professional models, they show a promising start to open-source fantasy football calculations.

When we step back from the data analysis, it is evident that the prediction of a NFL player's performance involves an infinite amount of factors that cannot be not tangibly measured by a single system. The quarterback may have gotten a flat tire on the way to the game and been more angry than usual, a stadiums light may be out and a receiver with poor eyesight may lose a hail mary in the dark, or a defensive lineman could have had a bad dinner the night before and is feeling a little bit sluggish compared to normal. There are factors nobody will know but the players themselves; hence there can never be a perfect fantasy football point predictor. Features such as injury status, team media coverage, bye week proximity, and stadium characteristics may all affect a player's performance in a week; those parameters are beyond the scope of this project but may be explored in future work.

Through our analysis of player and team statistics and prior fantasy football points, we were able to make several conclusions. First, week 1 data can be

approximated using a low rank model for data imputation. While imperfect, this method provides a start to providing richer pre-season predictions. Future work may use data from previous seasons or a player's college career for week 1 forecasts. In fact, this could be an area in which predictive modeling may be unfair to younger players (or worse players) for whom there is less data available. These players could be predicted to have fewer points or generally have higher variability. Fantasy football team owners may start a player who is predicted to perform well and if the player underperforms, the player may not be used again in the fantasy football lineup. While seemingly inconsequential, player popularity can be key to advertisement contracts or Pro Bowl selection.

While unlikely, there is some potential for these models to become a weapon of mass destruction through this channel. Fantasy football point predictions could affect performance: players may feel pressured to perform if they are predicted to have higher performance in a certain week. If teams begin to use these models, coaches may make decisions on which players start and how much of the game players play based on their predicted fantasy "productivity," which could change results. Players who are predicted to generate more fantasy points could play in more of the game, which allows more opportunities to be productive; the same is true of players who are predicted to generate fewer fantasy points. Usage of fantasy football predictors for actual football play may create negative feedback cycles for player performance.

Second, OLS is likely too simple for this project because of the sensitivity of quadratic loss functions to outliers. Players in the 72nd and above quantiles had disproportionate effects on the least squares model as found by the quantile regression modeling. The lasso model showed that the original feature matrix likely had too many features, and only a handful were useful for prediction purposes. In fact, the lasso model used a low number of features to generate predictions that had the lowest error.

With the immense revenue generated by the NFL and the high amount of Fantasy Football betting, we expected that there would be a plethora of articles on the back-end predictions of fantasy football points and player feature matrices. However, data science is still in its burgeoning stages for the NFL. Most articles in our search were about player style, team chemistry, and year to year performance. These descriptions are difficult to quantify and model, which leads us to believe that there are factors outside of the player statistics that could be significant enough to swing decisions on whether to play a player or not for a fantasy football league.

Thus, it may be interesting to use deeper features to investigate player abilities. One example could be to develop player chemistry ratings with teammates that benefit their scores by scoring correlations between player points. Along with this, we could also score how well offenses played against specific defenses to better rank team capabilities.

In conclusion, our model may not be the key to winning your next year's fantasy football league. But, we offer insightful machine algorithms that may vary from other black-box systems with human imputation, and the numbers from our methodology could serve as another piece of data when deciding a starting lineup and predicting player performance each week.