Project Midterm Report (Dream Team)

Darren Chang, Jack Vaughan, Zach Schaffer

1 Dataset Description

There are two 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 for offensive players each week. About 200-300 players are represented in each dataset for the 17 weeks of the regular NFL season. The three measures of fantasy football points for each player include the standard calculation (based on one point for ten yards of offense generated, and adds bonus points for touchdowns and subtracts points 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 half a point). Offensive statistics for each player is included, such as passing yards, rushing yards, and reception yards, completions, interceptions, and touchdowns. The overall quality of the data is great, as no player is missing information; however, players vary from week to week and there is little information at different levels of granularity.

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. These two datasets are joined together so that each player entry for each week has game data. There are $n = 52,744$ player-weeks represented in the data, with nominal, discrete, and continuous values.

2 Feature Engineering

For fantasy football projections, feature engineering is a critical part of transforming the data to be used in models. Projections must be created based on data from the previous weeks; for the first week, there is essentially no data. Two sets of features are engineered 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 a "baseline" factor for how well a player is performing throughout the season. Thus, there are 30 features for player statistics and one-hot encoded features for player team, opposing team, and player position.

This histogram 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 makes sense: 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 feature engineered data of the same base type. For example, `RushingYds` is highly correlated with `RushingYds_prev` and `RushingYds_cum`. Data that are sourced from the same position are also highly correlated with each other. 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).

Figure 1: Histograms of Previous and Cumulative Yardage

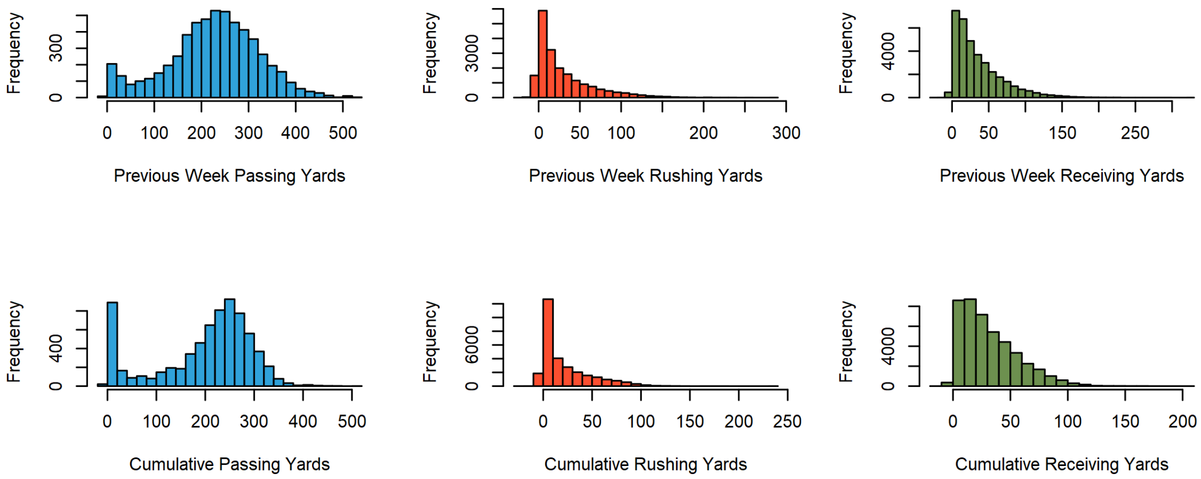
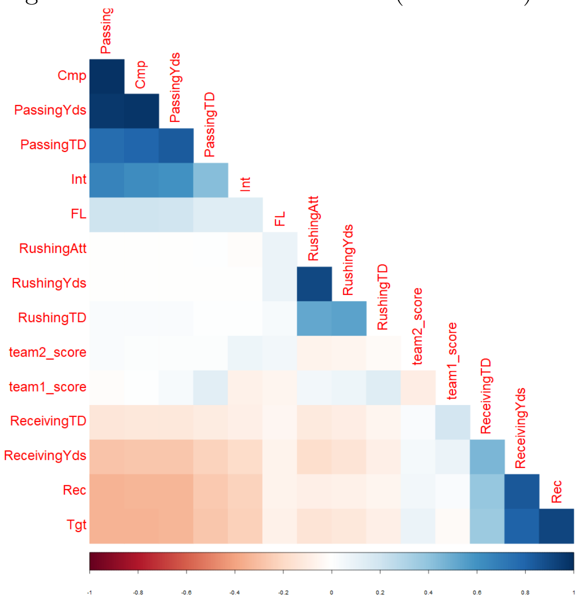


Figure 2: Correlation of Features (Truncated)

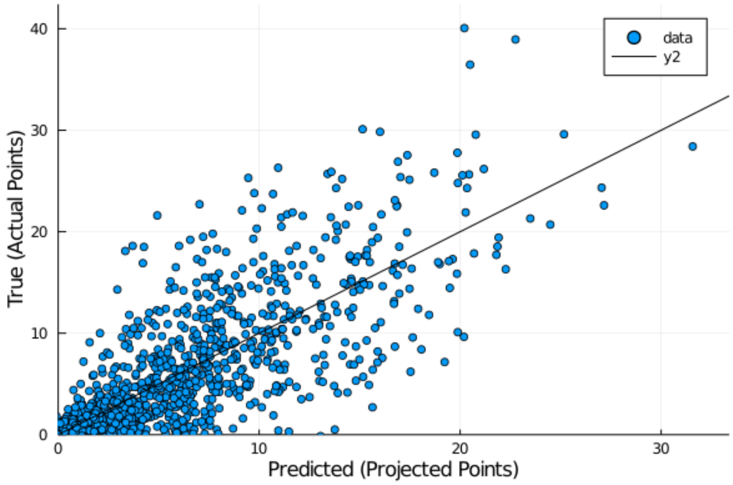


3 Preliminary Analysis

For preliminary analysis, we split the data into a training set with 80% of the data and a test set with the remainder. We pursued two models for this midterm report using the standard fantasy football points calculation as the output space: (1) the basic least squares l_2 regression and (2) lasso regression using the `GLM.jl` and `Lasso.jl` packages. Due to the high number of one-hot encoded variables (32 for the home team and 32 for the away team), there is high multicollinearity between the features, which results in an underdetermined OLS model. To solve this, we computed the OLS model while allowing rank deficient models and removed seven features that the rank deficient model indicated did not contribute to the model.

Then, we re-computed the base OLS model; the test MSE was 20.12. The scatter plot displays the predicted points versus the actual points, showing that the l_2 model may overfit to some of the outliers who have high predicted and actual points.

Figure 3: Scatter Plot of Predicted vs Expected Points (l_2 regression)



We also fit a lasso model onto the training set for the standard fantasy points, although have not yet used cross-validation to select λ. With $\lambda = 0.0013$, the lasso model uses 158 features to predict a test MSE was 19.87.

4 Plan

- We want to experiment with other loss functions that are intuitively useful for our data to better explore the bias-variance tradeoff between different models. For example, we think we should punish predictions that overestimate an individual's fantasy points compared to predictions that underestimate an individual's points. This is because overestimating points would lead to the wrong lineup for a fantasy matchup; underestimating points is simply an added bonus for the player. Thus, we could use a quantile loss function with $\alpha > 0.5$.
- We plan to use cross-validation methods to select hyperparameters (such as λ for the lasso regression model).
- One method of evaluating how effective our model was in predicting projected points is to run a simulation of a fantasy football league and compare if our projections resulted in better point totals for the selected starters. Fantasy Football Data Pros has simulation data for the 2019 NFL season from ESPN, which can be used to backtest our results.
- We hope to find data for defenses, which would make more model more complete.
- We plan to add more features to the model, including
  - More game-level features using the `game_id` field
  - The `nflWAR` framework as a measure of player contribution as a feature using play-by-play data, according to (Yurko, 2018)
  - Pre-game predictions from 538 and ESPN using the `nflfastR` package, as explained in this blog post.

  Because adding more features may create an overdetermined model, we will continue using lasso regression to better understand which features should be retained.