# Building an NFL performance metric

Seonghyun Paik (spaik1@stanford.edu)

December 16, 2016

## I. Introduction

In current pro sports, many statistical methods are applied to evaluate player's performance and ability. For example, WAR (wins above replacement) is a very popular measure of the player's performance in major league baseball (MLB). Especially, FWAR (Fangraphs WAR) for pitchers is very simple but powerful since it only uses 3 categories (Homerun, Strikeout and Based on balls) to measure performance of individual pitchers. Unlike baseball, which is easier as to build an individual performance metric due to nature of batter vs hitter match up, American football has more aspect of a team sports, and it's difficult and even can be misleading to evaluate performance based on individual statistics. However, it is still worthwhile to analyze which statistic category (such as passing vs. rushing, total passing yards vs completion rate) is more important to get more scores/win in a game since it can give information which aspect of game a team should fortify to improve their performance. Also, it will be an interesting to analyze how a current performance metric such as passer rate is reflecting to the actual team score.

In this project, I will focus on a supervised learning on building a performance metric which is directly related to the team's score, and finally a win. As inputs, all offensive and defensive statistic features except the direct way to get the score (touchdown, field goal, etc.) are included. Then I used different methods (linear regression, weighted regression, random forest and gradient boosting) to output a predicted score of the team and win/loss of a game.

## II. Previous Work

In the last year, a few groups worked on prediction and optimization of a fantasy football team. For example, P. Steenkiste used linear regression and random forest model to optimize a fantasy team [1] and P. Dolan et.al. used linear regression to predict quarterback performance in terms of fantasy points [2]. Although the target variables will be different for this project (game score of a game instead of fantasy points), both projects should deal with regression problems using similar inputs.

## III. Data Acquisition and feature selection

### 1. Data acquisition and processing

Individual NFL game statistics in 2013 is obtained from on the web [3]. Since each 32 teams will have 16 games in the regular season, the raw data contains data for 256 games. Two features are selected for my prediction variables - the scored points for each team and win/loss of the games. I've included additional 5 features (yard/attempt, pass completion percentage, etc.) since they could provide more intuitive analysis and traditional metric is also based on them.

From the 512 team score samples, I set aside 400 samples for training and validation, and the remainder (112 samples) for testing. For prediction of win/loss, I selected 200 game results for training and validation, and remaining 56 results are reserved for testing.

### 2. Feature selection

There were 28 features in the data set, but using all of the features can result in overfitting. I therefore explored the use of feature selection to shrink the number of input features. To select the most important features to keep, I ran sequential forward-based feature selection on the training examples.

| | |
|---|---|
| 1 | First Down |
| 2 | Passyard/Attempt |
| 3 | Turnover from opponent |
| 4 | Turnover given up |
| 5 | Rushyard/Attempt |
| 6 | Yard loss by sack |
| 7 | Penalty yard from opponent |
| 8 | Yard loss by penalty |

**Table 1**. *Top features for a team score*

Features were selected based on their mean squared error, using 5-fold-cross-validation. The features with top importance listed in the table 1, ordered by their importance. MSE from using different numbers of features are described in figure 1. It turns out using all of the features doesn't result in overfitting, but having similar performance comparing using top 8 features.

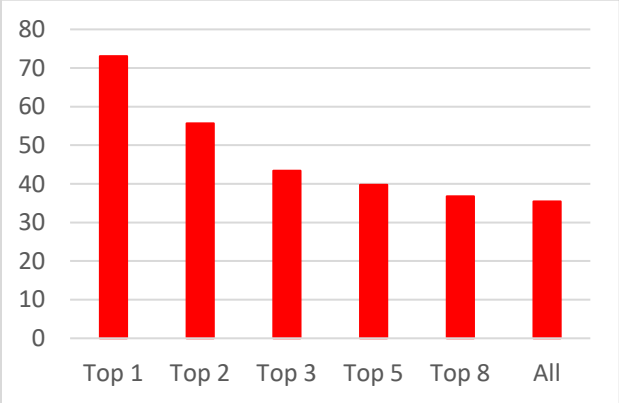


**Figure 1**. *MSE vs. number of features included*

## IV. Methods

### 1. Linear Regression

I suppose my learning task as a regression problem: given a processed list of features, I would like to predict scores of a team. Linear regression is a natural choice of baseline model for regression problems, so I first ran a linear regression using the top 8 features. To reduce the variance of the model, I first bootstrapped my training samples (make 100 cross validation sets by random sampling from the training data) and ran the linear regression model each time, and then used a weighted average, where the weights are inversely proportional to the model's cross validation error. The obtained model is

$$(\text{score}) = -12.09 + \sum_i c_i f_i$$

The coefficients and features are described in the table 2. MSE of this model is 36.7 points$^2$, equivalent to 6.05 point RMSE (root mean squared error).

| | $f_i$ | $c_i$ |
|---|---|---|
| 1 | First Down | 0.736 |
| 2 | Passyard/Attempt | 2.185 |
| 3 | Turnover from opponent | 2.629 |
| 4 | Turnover given up | -1.472 |
| 5 | Rushyard/Attempt | 1.229 |
| 6 | Yard loss by sack | -0.105 |
| 7 | Penalty yard from opponent | 0.037 |
| 8 | Yard loss by penalty | -0.022 |

**Table 2**. *Coefficients for the linear model*

### 2. Weighted linear regression

To capture local nonlinearities between the features and the score, I tried local weighting to the cost features for our linear regression model. Using the

Euclidean norm, my weight function for a training example x(i) with respect to an input example x was:

$$\text{w(i)} = \exp\left(-\frac{||x(i) - x||^2}{\tau^2}\right)$$

To make the Euclidean distance (the norm in the equation above) meaningful, I standardized features to zero mean and unit variance prior to computing the weights. The parameter $\tau$ in the weighting function is tried $0.5 < \tau < 100$, but I couldn't find a noticeable performance change in terms of MSE compared to the simple linear regression model.

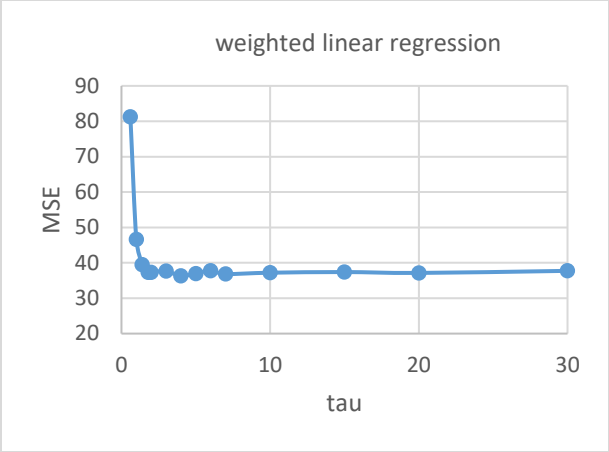


**Figure 2**. *MSE vs.τ in weighted linear model*

As described in figure 2, MSE is almost identical when $\tau \geq 2$, but trended to increase sharply as $\tau$ reduces when $\tau < 2$.

### 3. Random Forest

Random Forest method is one of the tree-based model. It builds a number of decision trees on bootstrapped training samples, but only consider a random sample of all possible predictors when split the data set so that each trees are decorrelated[4]. By taking average of the resulting trees, we can reduce the variance of the model and the model becomes more reliable.

An ensemble method that used a weighted sum of multiple random forest model, where each of the random forest models was fit using different parameter settings is used. The weights were inversely proportional to the models' cross-validation error. The MSE curve vs. number of the bootstrapped tree is plotted in the figure 3. As described in the figure, the MSE of the model is saturated at around 48, about 33% higher than the linear regression model.

### 4. Gradient boosting

Gradient boosting is another tree-based method, but unlike the random forest model, the trees are grown sequentially without using bootstrap sampling. In this model, we apply a slow learning rate and update the result weighted by the learning rate, and

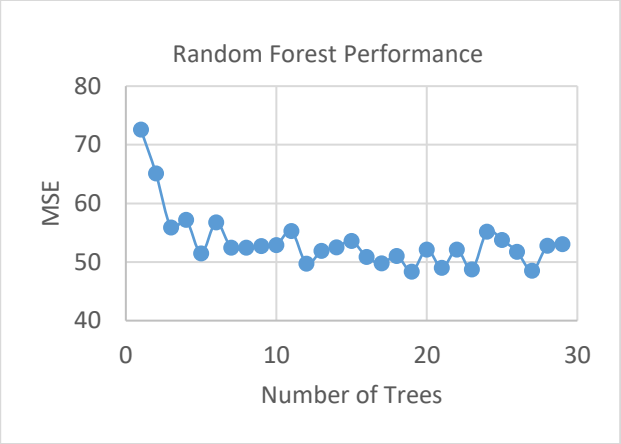


**Figure 3**. *MSE vs. number of trees in random forest model*

iteratively fit the decision tree to the residual from the model. Each of these trees are rather small with a few terminal node.

In this project, I ran a gradient boosting regression with tree depth of 1 and 2. Learning rate of 0.02 is applied. The MSE vs. number of iteration is plotted in the figure 4 below. The MSE is saturated after 500 runs of the iterations. The performance of the gradient boosting is comparable to the random forest method, about 33% higher than the linear regression model.

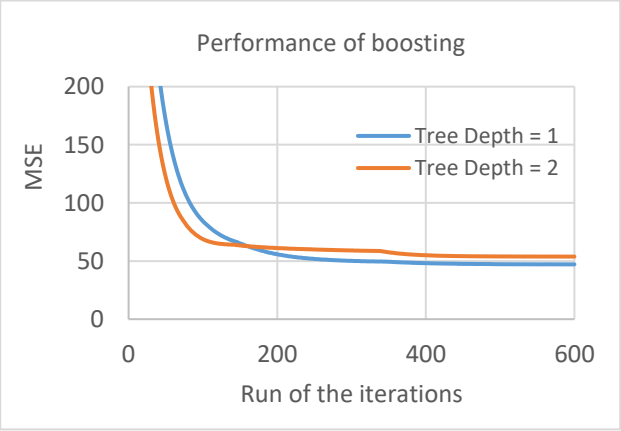


**Figure 4**. *MSE vs. number of iteration in boosting model*

## V. Results

Table 3 shows the test set performance of the optimized (with respect to the training set) model. The primary error metric is RMSE, which penalizes larger deviations superlinearly. For the best model, linear regression, the $R^2$ measure between predicted score and actual values in the test set was 0.6447.

| | MSE(point$^2$) | RMSE(point) |
|---|---|---|
| Linear Regression | 36.652 | 6.054 |
| Random Forest | 48.379 | 6.956 |
| Boosting | 47.14 | 6.866 |

**Table 3**. *Error for the scores across all models*

This model is also used to predict the outcome (win/loss) of the test samples based on calculated score prediction, and obtained 88.2% accuracy.

## VI. Discussion

### 1. Passer rate

As a popular traditional metric to evaluate performance of a quarterback, passer rating has a formula of [5]

$$(rate) = (PCT - 0.3) \times 83.3 + (YPA - 3) \times 4.167 + \left(\frac{TD}{ATT}\right) \times 333.3 - \left(\frac{INT}{ATT}\right) \times 416.7$$

A developed linear model from the training data, using the same features except the touch down rate in the passer rate is

$$(score) = (PCT) \times 4.92 + (YPA) \times 2.53 + \left(\frac{INT}{ATT}\right)_{opp} \times 55.3 - \left(\frac{INT}{ATT}\right) \times 60.4 + 3.45$$

Comparing two expressions, it turns out the passer rating overestimates the importance of pass completion percentage and intercept rate by factor of 10 and 2, compared to average pass yard. In other words, pass completion rate is not an important indicator to a team score, but has a large weight when evaluating the passer rate.

### 2. First down, pass vs rush

Number of the first down a team gets in a game is the strongest indicator of the score, but it's not included in the passer rate or fantasy point for running back or wide receiver. Instead, they include the number of touchdown scored, which is directly related to the score. However, when it comes to evaluate the performance of a player, touchdown is not a good metric since it does not credit previous plays which contribute to. Also current metrics credit a touchdown too significantly (equivalent to 80 yard in passer rate, 100 yard for quarter back and 60 yard for running back and wide receiver in fantasy point system.)

Comparison of average pass and rush yard per attempt showed that pass average per attempt is

much strong indicator related the team score. My model with 3 times bigger coefficient for the average pass yard, when all the coefficients are normalized to compare.

### 3. Building a new passer rate

Based on the discussion above, I suggest a new metric to evaluate quarterback performance using three most important features (first down, average pass yard and intercept rate)

$$(rate) = \left(\frac{firstdown}{ATT}\right) \times 31.5 + \left(\frac{PassYard}{ATT}\right) \times 2.07 - \left(\frac{INT}{ATT}\right) \times 110$$

## VII. Conclusion and future work

In this project, I focused on predicting NFL team score based on features not directly related to the score itself. Linear regression shows better estimate of the score than two tree based models, random forest and gradient boosting. The result of this project can be used to compare the importance of each NFL stat categories. It turns out that number of the first down, which is not included in either passer rate or fantasy point evaluation, is the strongest indicator on a team's score. On the other hand, pass completion rate, which is one of the four component

in evaluation of passer rate, affects very little on the team's score. From the developed model, the effect of 10% pass completion difference in scoring is equivalent to merely 0.2 yard/pass attempt.

As a future work, this model can be extended to actually predict the outcome of the future games. In order to do this, we will need to build a model which predict each stat categories based on the data of previous games of a team.

## References


[1] Paul Steenkiste (2015), "Finding the Optimal Fantasy Football Team", Stanford CS229 final report.

[2] Peter Dolan et.al.(2015), "Machine Learning for Daily Fantasy Football Quarterback Selection" CS229 final report

[3] https://www.statcrunch.com/app/index.php?dataid=1903988.

[4] Gareth James et.al., (2013) "An Introduction to Statistical Learning with Applications in R" Springer

[5]http://www.nfl.com/help/quarterbackratingformula