Identifying Over-Valued Players in Fantasy Football

AUTHOR: Michael Jermann **EMAIL:** MJermann@gmail.com

1. Introduction

This paper explores the application of machine learning algorithms to National Football League (NFL) statistics in order to predict fantasy football performance. Both supervised and non-supervised machine learning techniques are used in an effort to identify over-valued NFL players prior to the start of the NFL season, enabling fantasy football participants to reduce their risk of drafting players who underperform in the regular season.

1.1 Fantasy Football Overview

Fantasy sport participation has flourished in recent years. The Fantasy Sports Trade Association estimates that over 32 million people in the U.S. alone participated in fantasy football in 2012[1]. This is a huge industry that is continually growing as fantasy sports continue to permeate mainstream culture. For those unfamiliar, fantasy football team owners join leagues comprised of 8-12 people and participate in a draft, selecting real-life NFL players to join their fantasy football team. Throughout the season, owners compete by setting line-ups of the players on their team, earning points based on the statistics these players achieve in real-life NFL games that week. An owner wins if he/she scores more points than his/her opponent that week. Owners continually add new players and complete trades throughout the season, making it an extremely competitive and strategic activity.

1.2 High-Level Approach

The quantitative nature of fantasy football and the huge amount of freely-available data from outlets such as ESPN make it an excellent environment to apply machine learning. Existing predictive efforts are less algorithmic in nature and involve analyzing videos and scouting reports along with past statistics in order to forecast future performance. Prior to the season, hundreds of experts create player rankings meant to reflect the expected value of NFL players at each position, enabling fantasy football team owners to draft players more effectively. While these rankings may be accurate for most players, they are egregiously wrong for a good proportion each year. Furthermore, it these players that tend to truly impact an owner's success. Wasting early draft picks on players who significantly underperform is devastating to a fantasy football team, just as picking up an eventual top-performer near the end of the draft can save a season. Therefore, rather than apply machine learning to create better regression models like previous projects[2,3] , I have chosen to attempt to identify incorrectly valued players in the preseason. For the purpose of this project I focused on applying machine learning to identify the players that are over-valued in the pre-season (hereby referred to as "busts").

1.3 Assumptions and Scope

The player positions analyzed in this project are Quarterbacks (QBs), Running Backs (RBs), Wide Receivers (WRs), and Tight Ends (TEs). These positions were analyzed in separate training sets given that the statistics available for each varied. I operated under the assumption that the average fantasy football league consists of 10 participants who each start 1 QB, 2 RBs, 3 WRs, and 1 TE each week. Additionally, I calculated fantasy football points according to the ESPN Standard League Scoring Scale[4].

2. Preliminary Steps

Prior to applying machine learning, it was necessary for me to obtain the necessary NFL ranking and statistics, pre-process it into formats convenient for machine learning, and apply the "bust" criteria to it.

2.1 Data

In order to collect the necessary data, I made use of the Python web-scraping library BeautifulSoup[4]. There were three primary types of data required for this analysis:

1. **Preseason Rankings** – These are the aforementioned pre-season player rankings meant to predict the expected value of players. I obtained the ESPN "Top 200" player rankings[5] for 2008 through 2013 and converted them into positional rankings. On average this consisted of 26 QBs, 65 RBs, 65 WRs, and 20 TEs per season.

2. **Fantasy Football Statistics** – These statistics are the inputs to the fantasy football point function and determined how many points a player earned during the season. I obtained the data for 2008 through 2013 from the CBS Sports Website[6].

3. **Team and Player Statistics –** Various player and team statistics were obtained and normalized to act as the features for my training sets. Initially, only a player's statistics from the year immediately prior were considered. The data was obtained from www.pro-football-reference.com[7] spanning 2006 through 2013. This resulted in 51 features for QBs, 42 features for RBs, 37 features for WRs, and 37 features for TEs.

2.2 Bust Criteria

In order to use a classification algorithm, it was necessary to define a "bust" label for each training sample. I defined criteria based on a position's "Starting Threshold" – that is, the number of NFL players started per week in an average league (10 QBs, 20 RBs, 30

WRs, and 10 TEs). Basically, if the player was drafted to be a starter but ended up performing outside of that threshold, he is considered a bust. This is encapsulated in Equation 1 below.

Equation 1. **Bust Definition.**

$$START < (POST - PRE)$$

POST is the player's post-season integer positional ranking, PRE is the player's pre-season integer positional ranking, and START refers to the "Starting Threshold" which varies by position (QB: 10, RB: 20, WR: 30, TE: 10).

Occasionally players will be injured in the preseason and miss an entire NFL season, accruing zero fantasy points. Given that these players have achieved zero points they do not appear in the post-season positional rankings but should still constitute a bust. However, lowly ranked pre-season players who do not register statistics should not be considered busts. Therefore criteria was necessary for this case as well. If POST is undefined, the criteria used will be Equation 2 below.

Equation 2. **Bust Definition for DNP (Did Not Play) Players.**

$$START < (RANKLIMIT - PRE)$$

RANKLIMIT is the number of players ranked per position in the preseason (QB: 26, RB: 65, WR: 65, TE: 20) and START and PRE are defined as in Equation 1.

3. Analysis

Using the criteria established in section 2.2, I chose to use a classification algorithm to identify the "busts". I decided to use a Support Vector Machine (SVM) rather than the standard logistic classification algorithm given its increased accuracy and ease of parameter/kernel adjustment. The primary goal of classification was to obtain a list of players who, if avoid in the draft, decreases a participant's chances of drafting a "bust." Therefore, my primary concern was achieving a high recall along with low test error.

3.1 Support Vector Machines

In my initial analysis, I made use of the LibSVM Matlab Library[9] to perform Support Vector Machine classification for QBs, RBs, WRs, and TEs. A linear kernel was chosen because the training set already had a large number of features (as mentioned in section 2.1) relative to its size and high-order dimensionality did not seem necessary. The training set initially consisted of the player rankings and statistics from the 2008 through 2012 seasons (5 seasons) while the test set consisted of the player rankings and statistics from the 2013 season. At the time of analysis the 2013 NFL season was 70.79% complete, so player statistics were extrapolated to reflect the "projected" stats for the 2013 season. Table 1 below reflects the size of the training and testing sets by position along with the "bust" percentages.

Table 1. **Training Set and Testing Set Attributes**

	Training Set			Testing Set		
Position	Size	"Busts"	"Bust" Percentage	Size	"Busts"	"Bust" Percentage
QB	130	35	26.92%	24	4	16.67%
RB	325	93	28.62%	67	25	37.31%
WR	327	86	26.30%	67	17	25.37%
TE	97	30	30.93%	20	6	30.00%

3.2 Initial Results

Table 2 displays the error achieved by the initial SVM model on both the testing and training data. Additionally, test precision and recall by position is also present. The "N/A" indicates that for RBs, the model did not predict any busts.

Table 2. **Initial SVM Performance**

Position	Features	Train Error	Test Error	Test Precision	Test Recall
QB	51	19.23%	29.17%	28.57%	50.00%
RB	46	27.69%	37.31%	N/A	0.00%
WR	37	23.85%	29.85%	28.57%	11.76%
TE	37	20.62%	30.00%	50.00%	33.33%

The training and testing error was high for all four positions; however, the QB and TE models achieved acceptable levels of precision and recall. The precision and recall achieved by the WR model was acceptable as well; however, the RB model did not make any predictions and as such was extremely poor. The QB and TE training sets were much smaller than the RB and WR ones so it is possible this played a role in their better performance.

3.3 Training Data Analysis

Prior to further tuning the SVM models, I chose to analyze the training data in order to investigate is relevancy to the 2013 NFL statistics. Several recent NFL trends have fundamentally altered the strategies and statistics achieved by players (such as the 2010 rule change protecting wide receivers from helmet-to-helmet tackles[10] and the emergence of pass-catching tight-ends), and my suspicion was that older training data may have significantly lower relevance to the current 2013 season. Figures 1 and 2 below display the league-wide rushing/passing touchdowns and yards per season.

Figure 1. **Rushing and Passing TDs by Year**

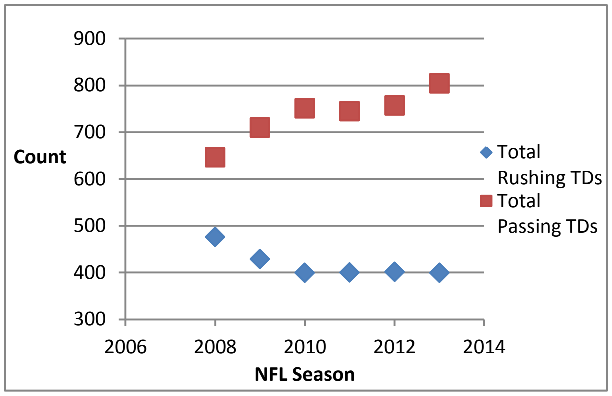
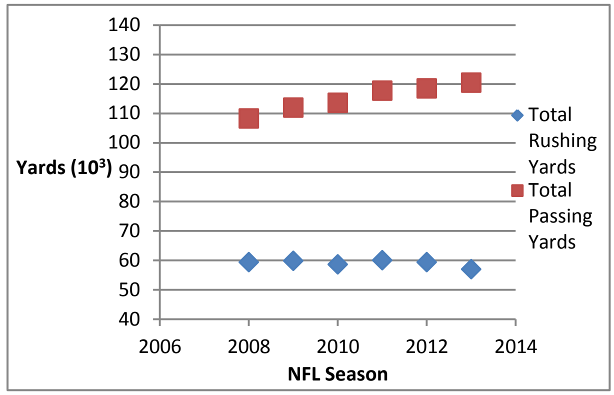


Figure 2. **Rushing and Passing Yards by Year**



There was a steep increase in passing touchdowns between 2008 and 2010, accompanied by a corresponding decrease in rushing touchdowns. Passing yards have also steadily increased between 2008 and 2013 while rushing yards have remained mostly consistent. This illustrates the increased emphasis on passing and receiving caused by offenses' focus on WRs and TEs at the expense of RBs. Therefore, it is likely that statistics from prior to 2010 provide little insight into the performance of today's players. In order to test this theory, I chose to omit the 2008 training data and retrain the SVM classification models. I kept the 2009 data present in order to maintain a sufficiently-large training set across all positions. The results of this are contained in Table 3 below.

Table 3. **SVM Performance with 2008 Data Removed**

	2008 Data Removed				Initial SVM			
Position	Train Error	Test Error	Test Precision	Test Recall	Train Error	Test Error	Test Precision	Test Recall
QB	22.12%	16.67%	N/A	0.00%	19.23%	29.17%	28.57%	50.00%
RB	25.57%	35.82%	100.00%	4.00%	27.69%	37.31%	N/A	0.00%
WR	23.94%	26.87%	42.86%	17.65%	23.85%	29.85%	28.57%	11.76%
TE	17.95%	30.00%	50.00%	33.33%	20.62%	30.00%	50.00%	33.33%

RB and WR display slightly lower test and train error but significantly improved precision. Recall is the most important metric for this problem in order to ensure our predicted "bust" list is as comprehensive as possible, and this metric increased for these positions as well. Therefore, I chose to remove the 2008 data from the RB and WR baseline SVM models but continue to use it for the QB and TE models, which did not experience performance improvement upon removing the 2008 season training data.

3.4 Error Analysis

The learning curves (training and testing error versus training set size) for the baseline models established in 3.3 are displayed in Figures 3-6 below.

Figure 3. **Tight-End Learning Curve**

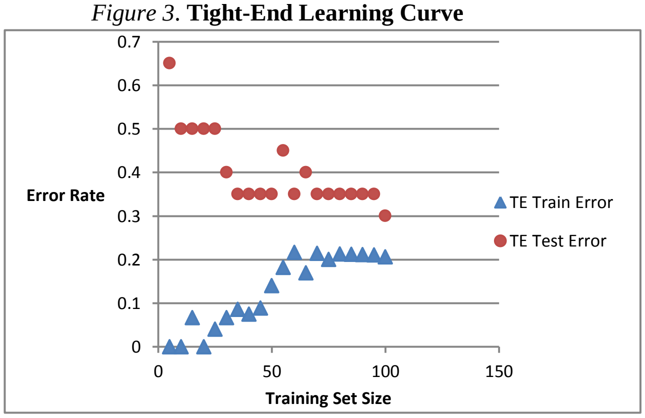


Figure 4. **Running Back Learning Curve**

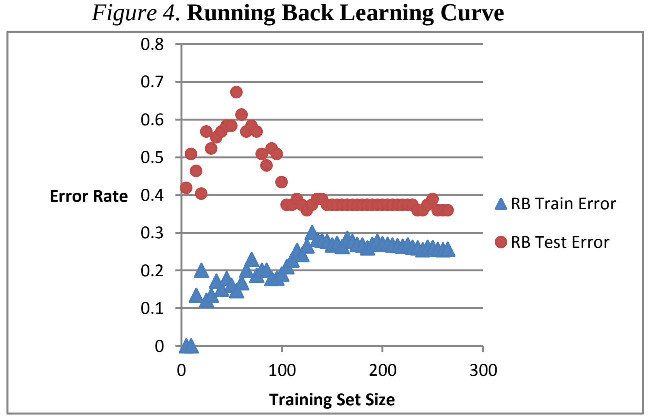


Figure 5. **Wide Receiver Learning Curve**

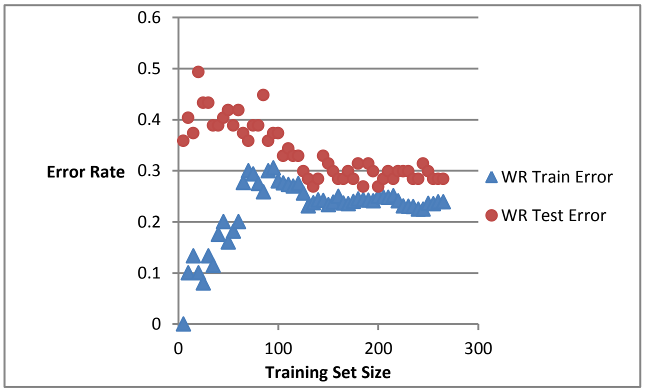
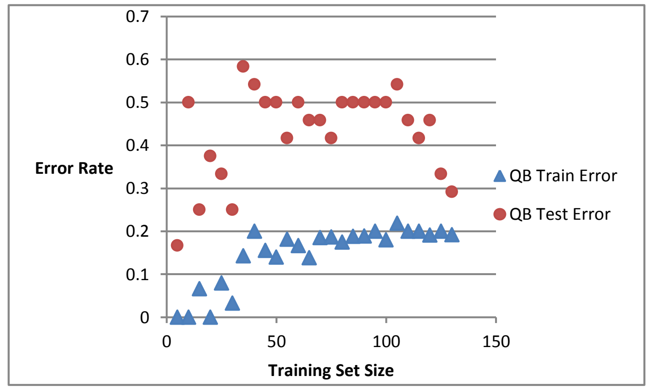


Figure 6. **Quarterback Learning Curve**



For TE, RB, and WR, the training and testing error appears to have plateaued and is not significantly improving with additional training set data. Additionally, the training error is high for these positions, indicating that these models have high bias. This implies that the current features used are not sufficient as to obtain an accurate model; rather, we need to add additional features. The QB testing error appears to be decreasing as the training set size increases, meaning that the accuracy may improve by reducing the number of features. Since the majority of the models displayed high bias, I chose to include each player's stats from two years prior as features in the training sets, the results of which are displayed in Table 4.

Table 4. **Classification accuracies for SVM on Test Dataset.**

	SVM Trained with Expanded Feature Set					**Baseline SVM**				
Position	Features	Train Error	Test Error	Test Precision	Test Recall	Features	Train Error	Test Error	Test Precision	Test Recall
QB	77	19.23%	41.67%	20.00%	50.00%	51	19.23%	29.17%	28.57%	50.00%
RB	67	23.66%	38.81%	33.33%	4.00%	46	25.57%	35.82%	100.00%	4.00%
WR	49	23.17%	26.87%	33.33%	5.88%	37	23.94%	28.36%	37.50%	17.65%
TE	49	19.23%	41.67%	20.00%	50.00%	37	20.62%	30.00%	50.00%	33.33%

For most models the training error decreased and testing error increased, indicating that over-fitting is present. This means that we most likely added too many features to the models. In order to determine the optimal number of training features for each model, I implemented forward filter search. For each model I continually iterated through all features, each time adding the one that most improved the testing error. Each time a feature was added, the test error and features used were recorded in order to identify the optimum feature set. I also implemented this separately with the goal of maximizing test recall rather than test error. Both results are displayed in Table 5.

Table 5. **Classification accuracies for SVM on Test Dataset.**

	Forward Filter Feature on Test Error					**Forward Filter Feature on Test Recall**				
Position	Features	Train Error	Test Error	Test Precision	Test Recall	Number of Features	Train Error	Test Error	Test Precision	Test Recall
QB	60	21.54%	16.67%	50.00%	75.00%	51	20.00%	20.83%	44.44%	100.00%
RB	46	25.57%	35.82%	100.00%	4.00%	59	24.43%	37.31%	50.00%	12.00%
WR	40	22.39%	23.88%	66.67%	11.77%	43	23.55%	34.33%	25.00%	17.65%
TE	38	23.08%	25.00%	100.00%	16.67%	39	21.79%	40.00%	33.33%	33.33%

Forward filtering features on testing error increased the precision for all models except for RB; however, recall decreased for WR and TE. Forward filtering features on test recall dramatically improved model performance for QBs and RBs but increased the test error rate of WRs and TEs, along with decreasing their precision.

3.5 Cluster Analysis

In order to gain additional insight into the data I chose to implement k-mean clustering without using my pre-defined “bust” criteria. For each training set, I used two clusters and iterated through all combinations of two features. Each time a cluster contained at least 10% of the training set, I calculated the average position “over-value” – that is, how much each of the players in that cluster were over-valued in the pre-season on average. My pre-conceived notion of a bust may have limited my models initially so analyzing clusters with high positional average positional “over-values” could provide insight into important features along with what attributes should constitute a playera “bust”. Table 6 contains a sample of the highest average “over-value” clusters by position.

Table 6. **Highest Average “Over-Value” Clusters by Position**

Position	Feature 1	Feature 2	Cluster Size	Average Positional “Over-Value”
QB	High Times Sacked	Low Passing TD	19	7.58
RB	High Rushing Attempts	High Fumbles	34	16.56
WR	High Receptions per Game	High Passing Yards (Team)	34	20
TE	Low Points (Team)	Low Rushing Yards (Team)	15	10.67

An interesting insight from the WR cluster is that a wide receiver with a high reception total on an offensive-minded team appears likely to not replicate the results the next year. Additionally, a RB who receives a high volume of carries and often fumbles is at risk to be a “bust” the next year as well, as displayed in Table 6.

4. Conclusions

The optimal models I was able to achieve for each position are contained in Table 7. “Expected ‘Bust’ Percentage” refers to the probability that a fantasy football player using these models will draft a “bust” at each position.

Table 7. **Optimal Classification Models.**

	Optimal Classification Models					Forward Filter Feature on Test Recall			
Position	Features	Train Error	Test Error	Test Precision	Test Recall	Size	“Busts”	“Bust” Percentage	Expected “Bust” Percentage
QB	51	20.00%	20.83%	44.44%	100.00%	24	4	16.67%	0.00%
RB	59	24.43%	37.31%	50.00%	12.00%	67	25	37.31%	36.07%
WR	37	23.94%	26.87%	42.86%	17.65%	67	17	25.37%	23.33%
TE	39	21.79%	40.00%	33.33%	33.33%	20	6	30.00%	28.57%

These optimal classification models provided significant advantage in avoiding busts for QBs (as compared to the actual “bust” percentage). They also provided slight advantages for RBs, WRs, and TEs. Next steps include expanding upon clustering analysis in section 3.5 to explore alternative “bust” definitions and gain insight into additional impactful features. Furthermore, expanding this analysis to identify “under-valued” as well as “over-valued” provide additional value and consist of a similar analysis.

References

[1]Fantasy Sports Trade Association. *Fantasy Sports Trade Association*. November 15, 2013, http://www.fsta.org/industry_demographics.

[2]Predicting Fantasy Football Performance with Machine Learning Techniques. Nitin Kapania. [pdf]

[3]Predicting Fantasy Football Results. Matt Bookman. [pdf]

[4]http://games.espn.go.com/ffl/resources/help/content?name=scoring-formats

[5]Beautiful Soup Python Web-Scraping Library (http://www.crummy.com/software/BeautifulSoup/)

[6]ESPN 2008-2013 Top 200 Rankings (http://sports.espn.go.com/fantasy/football/ffl/story?page=nfl2k8DKtop200)

[7]CBS Fantasy Football Statistics (http://fantasynews.cbssports.com/fantasyfootball/stats)

[8]Pro-Football-Reference.com (http://www.pro-football-reference.com/)

[9]Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2:27:1 –27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~sjlin/libsvm

[10]http://www.nfl.com/news/story/09000d5d81990bdf/article/new-nfl-rules-designed-to-limit-head-injuries