Projecting a Quarterback's Fantasy Football Point Output for Daily Fantasy Sports using Statistical Models

Nicholas King
University of Texas at Arlington
701 S. Nedderman Drive
Arlington, TX 76019

Aera LeBoulluec, PhD
University of Texas at Arlington
701 S. Nedderman Drive
Arlington, TX 76019

ABSTRACT

In the new age of daily fantasy sports (DFS), fantasy football has become an enormous revenue generator for DFS sites, such as DraftKings and FanDuel. Both companies are valued over $1 billion. However, previous analysis done by popular DFS site Rotogrinders, has shown that only the top players are consistently winning, the top 10 players much more frequently than the remaining 20,000 players. Using complex statistical models they're able to identify top athletes and value picks (based on an athlete's draft 'salary') that the average player might not be aware of. There is a need to evaluate which methods and algorithms are best at predicting fantasy football point output. These methods could then be applied to future DFS contests to see if they can predict other fantasy sports as well. There are few resources available on this subject, as DFS are still relatively new and few people publish their work, since they generally develop these models for their own financial gain. This research will attempt to find some effective statistical models to predict the weekly fantasy point output of a quarterback.

General Terms

Predictive Analytics, Statistical Modeling, Algorithms, Regressions.

Keywords

Fantasy football, daily fantasy sports, statistical models, machine learning, predictive analytics, DraftKings, FanDuel.

1. INTRODUCTION

Daily Fantasy Sports are still a somewhat new phenomenon. The two largest companies in the DFS world are DraftKings and FanDuel, together controlling around 95% of the DFS market [1]. It's important to understand the difference between traditional fantasy football and the variant found at DFS sites. In the traditional format a league consists of 10-12 teams, each run by a member of the league. Before the NFL season starts, a draft is held by the league members to fill out their team's roster. Usually this roster is composed of one quarterback (QB), two running backs (RB), two wide receivers (WR), one tight end (TE), a 'flex' spot (where a RB, WR, or TE may be started), a defensive/special teams unit, and a kicker. You're then able to draft and assemble your 'bench' players as you see fit. Each week in the league two players are matched up against each other. The fantasy points your team scores are a direct reflection of how your individual athletes do in their actual NFL games.

The length of the competition is a significant difference between DFS football and the traditional format. In the latter, a league's competition lasts nearly the length of an actual NFL season. The NFL regular season lasts for 17 weeks, fantasy football usually lasts 13-14 weeks, with 2 weeks of playoffs following. A winner is determined before the NFL playoffs begin. DFS sites rarely have these season-long commitments. They are an accelerated form, with most competitions taking place over the course of a single day or week. DFS allows you to draft a new team each week and eliminates the other responsibilities of season-long commitments, such as trading, dropping or adding athletes, or having to manage a bottom-feeding team all season, which keeps players engaged.

DFS football is also more money driven. DFS websites make their living from making fantasy football a cash game. Each week you pay to enter a team, be it from a couple of dollars, to several thousands, depending on the type of league and competition you'd like to face. DFS sites receive a portion of the entry fee from each player and then pay out the rest to the winner. While the traditional format pits one member's teams against another in a head-to-head battle, DFS sites offer you the chance to play in nationwide contests where the highest scoring roster takes the winnings among all entries in that league. This has led to large payouts for many members.

1.1 The Benefits of Statistical Models

DFS sites also add a salary component into the drafting process. Traditional fantasy sports allow you to draft the best athlete possible that is still on the board when it's your turn. DFS sites allocate participants a fixed salary cap of $50,000 that must be used to draft an entire roster. This means you not only have to pick and choose which athletes you think will do well, but also athletes that you can afford. This puts a premium on finding 'sleepers' or value picks. For example, an elite-level quarterback with an easy matchup might cost $9,000, and a mid-level QB with an average matchup might cost $6,500. If you choose the elite QB you've now spent nearly 20% of your salary cap on one player and still need to draft additional athletes (2 RBs, 2 WRs, 1 TE, 1 FLEX, 1 D/ST, 1 K) to fill out your team. So if you can find a QB, or any position for that matter, who is relatively cheap, but will give the same point output as an expensive QB, you've already positioned yourself to do better since you'll have more money to spend on other high-level athletes.

This is how the top players with statistical models have done so well. Previous analysis has shown that the top 10 players enter hundreds of lineups a day and win an average of 873 times *daily*. The remaining field of about 20,000 players wins just 13 times per day [2]. These top players are able to generate hundreds of different lineups where they try to optimize the relationship between projected QB point output and cost. Finding those sleeper picks also benefits them in the nation-wide contests even more. Picking athletes that have a low ownership percentage can provide separation between your entries and thousands of others. For example, if you and 20% of the nationwide participants start an elite quarterback like Aaron Rodgers, it won't give your team a lot of separation if he does well, since so many other people also started him. But if you started Andy Dalton, whom just 2% of entries started, and he does well, then you've just potentially passed up thousands of entries that didn't pick him - and you saved roster money doing so. Spotting a value pick by Dalton allowed you to have more money to spend on other positions than if you had paid more for Aaron Rodgers. The research presented here will seek to identify variables and models that can accurately predict the point output for QBs. This could then allow one to locate athletes that will offer solid point output at a reduced cost - allowing for more money on better athletes at other positions. This methodology can be used in the future to project other positions.

It's also important to briefly point out how challenging projecting weekly point output has been and remains. Almost all websites that formulate fantasy football projections refuse to keep their projections up past the current week. This is for a simple reason - they don't want visitors or paying subscribers to see how inaccurate their projections might have been. Finding data on historical weekly projections is challenging because of this. Fig. 1 shows how hard it is to project an athlete's, or in our case, a quarterback's output on a week-to-week basis.

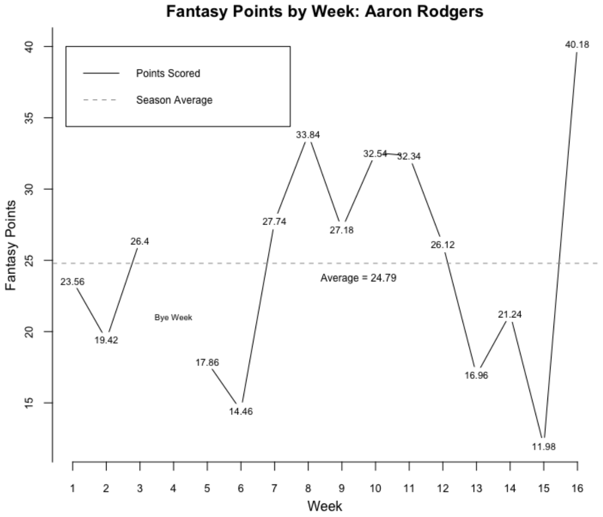

Fig 1: Aaron Rodgers' DraftKings point output by week.

2. DATA AND METHODOLOGY

The data used in this research was scraped together from several websites and sources [3]-[6]. Weekly QB performances from weeks 1-7 of the 2016 NFL season were used as a training dataset, while weeks 13-16 were used as a testing dataset. The training set led to a study of 43 different quarterbacks and 212 games played. Thus this dataset had 212 samples. The training dataset featured 33 QBs, with 126 samples. The training and testing sets were reduced to focus only on QBs that were starters. This is logical, as a player using DraftKings would never select a second or third string QB, despite several of these QBs logging statistics during games due to a starter's injuries. Imputation was done using column means where necessary. This was the case when working with rookie quarterbacks. Many of the variables in the datasets were values corresponding to performance metrics from the previous season. As a rookie, of course, you don't have a previous season's results. Imputing these missing values with the column mean is a typical practice done in data mining and allowed us to work with a full dataset in the research. Our final training and testing datasets evaluated over 50 different attributes, including a QB's height and weight, years in the league, NFL combine test results (such as 40-yard-dash time and Wonderlic test scores), red zone touchdowns, previous season's QB rating and pass attempts, and their weekly DraftKings salary, among many others. The dependent variable was the weekly fantasy points scored by the quarterback in a standard DraftKings league, *DK_points_scored*. A quarterback in a DraftKings league accumulates points as shown in Table 1.

Table 1. DraftKings Scoring Summary

Offensive Play	Points Awarded
Passing Touchdown (TD)	+4
25 Passing Yards	+1 (+0.04 pt/per yd)
300+ Yard Passing Game	+3
Interception	-1
10 Rushing Yards	+1 (+0.1 pt/per yd)
Rushing TD	+6
100+ Yard Rushing Game	+3
10 Receiving Yards	+1 (+0.1 pt/per yd)
Reception	+1
Receiving TD	+6
100+ Yard Receiving Game	+3
Fumble Lost	-1
2 Point Conversion (Pass/Run/Catch)	+2
Offensive Fumble Recovery TD	+6

Therefore, a QB who threw for 325 yards, 2 TDs, and 1 interception while rushing for 10 yards would be expected to score 24 points in a standard league [7].

The software used to analyze the data is R, a programming language for statistical computing and graphics. The models evaluated in this paper are Random Forests, Boosting, Principal Component Analysis, and Support Vector Regression. The R software includes packages that are able to formulate these models.

Each statistical model was trained on the training dataset and evaluated on the unseen testing dataset. The accuracy was calculated by finding the root mean squared error (RMSE) and the mean absolute error (MAE) of the model on the testing dataset. A comparison of the models will be shown in results section later. To establish a baseline a 'best guess' model was determined. Here the average QB point output of the training data was compared to the points scored of the test data and the RMSE and MAE found. A model that produced less accurate results than the best guess would be undesirable. In the following section the highest-performing individual models that were tested will be introduced and summarized.

3. MODELS AND METHODS

3.1 Tree-Based Methods

3.1.1 Random Forests

Random forests are an ensemble of different regression trees and are commonly used for nonlinear multiple regression. Regression trees are frequently used in data mining to create a model that predicts a continuous variable based on the values of numerous independent variables. The most popular way to do this is through the use of the Classification and Regression Trees (CART) decision tree methodology. The CART methodology was introduced in 1984 by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone as an umbrella term to refer to classification and regression trees [9]. The target variable in regression trees is continuous and the tree is used to predict its value. Regression trees have numerous advantages such as their interpretability and their ease of explanation, but generally do not offer the level of accuracy that most other regression approaches can offer. Methods that aggregate many regression trees, however, can offer more accurate results. Random forests are one of these more accurate models. This method constructs hundreds or thousands of multiple singular regression trees and then outputs the mean prediction of the individual trees. This method is much more powerful than a basic regression tree.

The model is fitted to the target variable using all of the independent variables. For each independent variable the data is split at various points. At each point the sum of squared error (SSE) is calculated between the predicted value and the actual value. Then the variable resulting in the minimum SSE is selected as a node to split on [9]. This process is continued until the entire dataset is covered.

In the context of decision trees, bootstrap aggregation, or bagging, is frequently used. Bagging is a general procedure for reducing the variance of a statistical learning method. The key to bagging is that trees are repeatedly fitted to 'bootstrapped' subsets of observations. Here we bootstrap by taking repeated, random samples from the training dataset. Therefore, a number of different bootstrapped training datasets are generated. In the context of regression trees this means that *B* regression trees are constructed using *B* bootstrapped training sets. Then we average out the resulting predictions, which reduces the variance [10]. In our work 500 trees are combined into this single procedure.

Bagging was used in our work with random forests. When building decision trees for random forests a random sample of *m* predictors are chosen as split candidates from the full set of *p* predictors. Typically *m* is only 1/3 of the predictor variables, *p*. The main difference between bagging and a standard random forests model is the choice of predictor subset size *m*. If *m* = *p*, then this simply amounts to bagging. After running the random forests method on our dataset we were able to output some variable importance plots and calculate the RMSE and MAE metrics. Fig. 2 shows several variables in the model and two different measures of variable importance. The first measures the mean decrease of accuracy in predictions when a given variable is excluded from the model. The second measures the total decrease in node impurity that results from splits over that variable. These results indicate that the two most important variables are the Average Projected points from all sources considered and FantasyPros' projected points. It's also interesting to note that the previous week's points scored was not considered very important. One might expect that if a player had put up several consistent performances in a row that this would be a good indicator of future success, but as our earlier example of Aaron Rodgers' weeks 15 and 16 show, this is not necessarily the case.

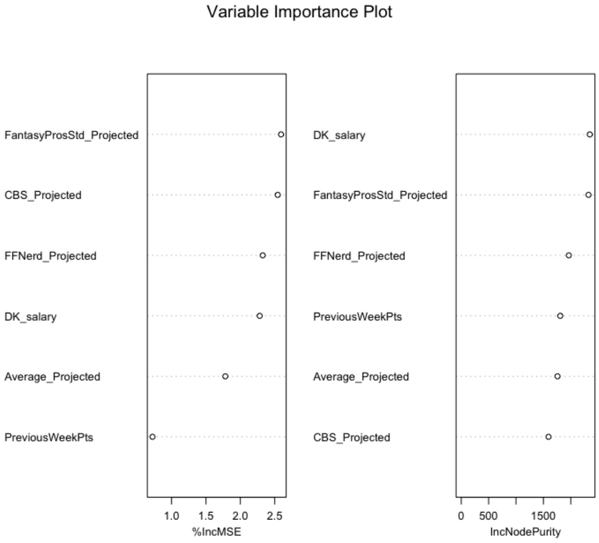

Fig 2: The importance of individual variables in the Random Forest.

3.1.2 Boosting

Another tree-based method that was tested was *boosting* - an additional approach for predictions resulting from a decision tree. Like bagging, boosting is a general approach that can be applied to many different methods for regression. Bagging involved creating multiple copies of the original dataset using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model. Each tree was independent and grown on a bootstrap dataset. Boosting works similarly, but the trees are grown sequentially, using information from previously grown trees. Boosting also doesn't involve sampling like bagging does; each tree is fit on a modified version of the original dataset. The idea behind boosting is to fit decision trees to the residuals from the model, rather than the outcome variable of *DK_points_scored*. Each new decision tree is added into the fitted function in order to update the residuals from the model. Since we are addressing the residuals, the model slowly improves in the areas where it does not perform well. This algorithm is considered a *slow learner* in the data science world, as it gradually improves the model, offering small improvements in the residuals. Typically, slow learning models perform well.

Boosting and bagging also differ in that the construction of each tree in boosting depends strongly on the trees that have already been grown [10]. In R the boosting algorithm is run with the gbm package, which allows us the option to set parameters for the distribution and the number of trees to sequentially grow. In our research we found the best results growing 5000 trees.

3.2 Principal Component Analysis

Our data has a large set of variables, which made a Principal Components Analysis (PCA), or Principal Components Regression (PCR), a logical method to explore. A PCA is often used to obtain a low-dimensional set of features form a large number of variables. A PCA models the variation in a set of variables in terms of a smaller number of independent linear combinations (principal components) of those variables [11]. The analysis refers to the process by which principal components are computed, and the subsequent use of these components in understanding the data. When faced with a large set of correlated variables, as our data reflects, principal components allow us to summarize the set with a smaller number of representative variables that collectively explain most of the variability in the original set. PCA is considered an 'unsupervised' approach, since it involves only a set of features $X_1, X_2, \ldots, X_p$, and no associated response variable Y [10].

The principal components regression (PCR) approach involves constructing the first M principal components, $Z_1, \ldots, Z_M$, and then taking these components and using them as the predictors in a linear regression model that is fit using the least squares method (least squares regression is a linear fit of a regression line that has the smallest possible value for the sum of the squares of the residuals). The key idea is that often a small number of principal components suffices to explain most of the variability in the data, as well as the relationship with the response. In other words, we assume that the directions in which $X_1, \ldots, X_p$ show the most variation are the directions that are associated with Y. This assumption is not always guaranteed, but is reasonable enough to provide good results. If we are to assume that this is true, then fitting a least squares model to $Z_1, \ldots, Z_M$ will lead to better results than fitting a least squares model to all our predictors $X_1, \ldots, X_p$), since nearly all of the information contained in the predictors is already present in the principal components [10]. If you were to use $M = p$, in which the number of principal components were equal to the number of predictors, then you would simply be performing a least squares regression. Thus one can begin to see how the PCA/PCR method makes effective use of reducing the dimensionality of the data. Reducing the number of predictors and principal components ended up improving our accuracy metrics.

Once all of the principal components have been computed, they can be plotted against each other in order to produce low-dimensional views of the data as shown in Fig. 3. It's important to note, however, that while the PCA/PCR method was one of the most accurate methods we employed, it is a dimensionality reduction method and *not* a feature selection method such as a random forest or standard multiple linear regression. This is because each of the M principal components used in the regression is a linear combination of all p of the *original features* [10].

It is necessary to perform the PCR only after standardizing each variable. This ensures that all variables are on the same scale. The PCR was run on our dataset, with the ideal number of principal components shown to be four. Fig. 3 above represents the principal component scores for the first two components and the loading vectors in a single 'biplot' display.

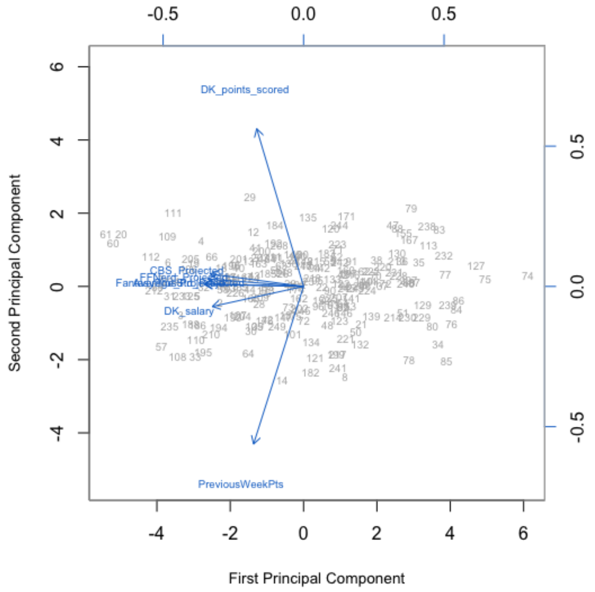

Fig 3: Depicting the first two principal components of the quarterback data.

Fig. 3 shows that the first loading vector places approximately equal weight on the variables for several different websites' projections, a quarterback's salary on DraftKings, and a QB's Average Projected points, and much less weight on Previous Week Points. This makes sense when considering Fig. 2 from the Random Forest. That model also determined that points from the previous week were a poor predictor of current week output. Variables that are found close together in this plot indicate that they are correlated with each other. As with the random forest method, the accuracy of the model's output was measured by computing the RMSE and the MAE.

3.3 Support Vector Regression

Finally, another unique form of regression, support vector regression, (SVR) was studied. SVR is based off of the same principles as the popular classification algorithm support vector machine (SVM), but with a few small differences. SVM is a classifier based on a hyperplane that does not perfectly separate the classes, but does give greater robustness to individual observations [10]. The SVM classifier allows some observations to be on the wrong side of the hyperplane. Since our research doesn't focus on predicting a class, but instead on outputting a continuous value, it is hard to predict the information at hand, which has infinite possibilities. Therefore, it allows for an error term, *epsilon*, which forms boundaries of the regression line [12]. Any errors greater than the epsilon threshold are penalized, which improves the SVR model's accuracy. SVR also tries to reduce model complexity. The main ideas between the SVM and the SVR are the same, however: to minimize the error and individualize the hyperplane that maximizes the margin.

Fig. 4 shows the idea of the SVR. The regression line is in the middle, bounded by the epsilon lines, with the acceptable data points inside. The support vectors are the points that are found on the boundary lines. Using the `e1071` package in R and its associated `svm()` function, we were able to run the SVR and fine tune *epsilon* value that produced the smallest root mean squared error (RMSE) and mean absolute error (MAE) values.

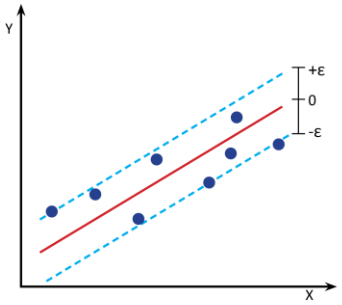

Fig 4: SVR shown with the margin of tolerance (epsilon).

4. RESULTS

The table shown below gives a summary and comparison of the results for each model tested. The Principal Component Regression is far better than the rest of the competition, with lower root mean squared error and mean absolute error values than any other method.

Table 2. Model Comparison

Rank	Method	RMSE	MAE
1	Principal Components Reg.	4.24	3.20
2	Support Vector Reg.	7.30	5.62
3	Boosting	7.70	6.09
4	Random Forest	7.72	6.10
5	Baseline/Best Guess	7.80	6.19

It is also important to note, though, that all of the models investigated here gave better results than the 'best guess' model that was based strictly on the average value of the response variable *DK_points_scored*.

We also compared our results to a leading popular sports website, CBS Sports, which offers fantasy football projections. Fig. 5 shows that our preliminary model is already more accurate than CBS Sports, which has a RMSE and MAE of 7.32 and 5.80, respectively. This is important because it reflects that our model has the potential to output more effective results than other mainstream offerings. The popularity and reach of websites like CBS Sports and ESPN is undeniable. There's no doubt that many of the participants in DFS competitions consider their projections when drafting a team. If our model is able to output more accurate results then we are already getting a leg up on the competition.

5. CONCLUSION

It is encouraging to see favorable results with a variety of different algorithms. The next step would be to try to fine-tune these models more in an attempt to further reduce the RMSE and MAE metrics and explain more of the unknown variation. Exploring the effect of injuries and how to handle them in the data might also yield better results. Some quarterback's point projections were way off because they left the game early with injuries. Other interactions that could be further researched and quantified might be the effect certain stadiums or teams have had on a quarterback in the past, or how strong of a role a coach, offensive coordinator, or opposing defensive coordinator and his typical scheme appears to play in point output. Of course like any model, a larger dataset only provides more accurate insights, so another year's worth of data would be enormously helpful.

It will also be interesting to see if the PCA method works as well at projecting other fantasy positions, such as running back and wide receiver. Oftentimes these positions can have even higher variability than the quarterback position. Once the correct models are identified to model individual positions on a team this research will focus more on the optimization issue of expect point output versus costs. This will ensure our model is able to generate a team that offers the most 'bang for the buck.'

Finally, this study would appear to support the idea that daily fantasy football is in fact a game of skill and not luck. With a well-developed, robust statistical model, a player seems to have an advantage over individuals using their best guess, or those utilizing the projections off of a popular free sports website.

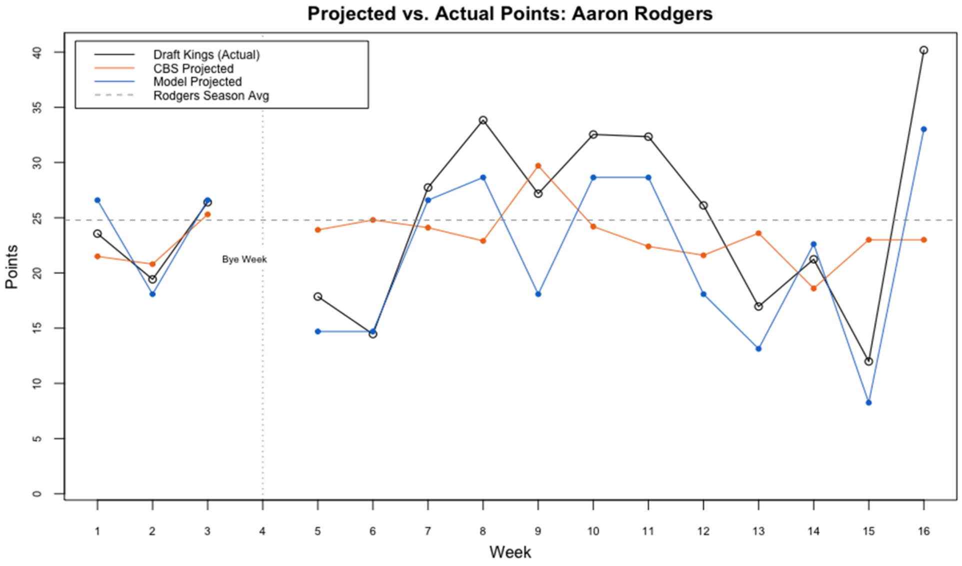

Fig 5: Aaron Rodgers' actual DraftKings' points compared to projections from CBS Sports and the PCA model developed in this research

6. REFERENCES

[1] Grossman, Evan. "Draft Kings and FanDuel Announce Merger." NY Daily News. New York Daily News, 18 Nov. 2016. Web. 09 Dec. 2016.

[2] Brustein, Joshua, and Ira Boudway. "You Aren't Good Enough to Win Money Playing Daily Fantasy Football." Bloomberg.com. Bloomberg, 10 Sept. 2015. Web. 12 Oct. 2016.

[3] "Fantasy Football Nerd - Fantasy Football Rankings." Fantasy Football Rankings: Quarterbacks. Fantasy Football Nerd, n.d. Web. Jan. 2017.

[4] "Fantasy Football Projections." CBSSports.com. CBS Sports Fantasy, n.d. Web. Jan. 2017.

[5] "Fantasy Football Projections." QB Projections - Consensus Fantasy Football Stats for Quarterbacks. FantasyPros, Sept. 2016. Web. Jan. 2017.

[6] "NFL Fantasy Football Stats." NFL Fantasy Football Stats & League Leaders. Fantasy Data, Sept. 2016. Web. Jan. 2017.

[7] "Fantasy Football Contest Rules & Scoring." DraftKings - Daily Fantasy Sports for Cash. DraftKings, n.d. Web. 29 Dec. 2016.

[8] Rao, Venky. "Introduction to Classification & Regression Trees (CART)." Data Science Central. N.p., 13 Jan. 2013. Web. 29 Dec. 2016.

[9] Sharma, Ankit. "How Does Random Forest Work for Regression?" Quora. N.p., 19 Aug. 2014. Web. 22 Dec. 2016.

[10] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. "Introduction to Statistical Learning." Introduction to Statistical Learning. Springer, 2013. Oct. 2016.

[11] "Overview of Principal Component Analysis." Overview of Principal Component Analysis. SAS Statistical Discovery, n.d. Web. 10 Jan. 2017.

[12] Kowalczyk, Alexandre. "Support Vector Regression with R." SVM Tutorial. N.p., 19 Jan. 2016. Web. 3 Jan. 2017.

[13] Sayad, Saed. "Support Vector Regression." Support Vector Regression. N.p., n.d. Web. 2 Jan. 2017.