**Understanding the NFL**

**June 6, 2013**

**Patrick McNamara**

**Abstract**

Baseball may still being America’s pastime, but the NFL is our present and probably near-future favorite sport. Yet compared to baseball or even basketball, little is understood about a sport that relies on complex play-formations and ‘gut instinct’. By analyzing play-by-play data over the last 11 seasons, I provide high-level insights on what impacts the outcome of games, where some of our basic assumptions about the game hold true and where they are unfounded. While the model is less than reliable at predicting total points per game, it has proven more valuable in predicting whether or not the home team wins.

**Introduction**

The NFL is America’s most valuable sport in terms of revenue, and has been for quite some time. League income has grown from $8.5 billion in 2010 to $9.5 billion in 2013 and the commissioner recently set a goal of $25 billion in revenue by 2027 (Kaplan, 2013). Teams are spending $120M+ per year on player salaries (Associated Press, 2013) and have dozens of scouts working the draft to pick the best personnel possible. However when it comes to analyzing on-field data, the sport is left lacking, especially compared to baseball (Bradley, 2013). NFL teams don’t necessarily know (or at least talk publicly about) the value of a marginal rush vs. a marginal pass, the effect of travel on win probabilities or whether the ‘spotlight’ of a primetime game changes everything or has no effect.

**Methods**

I had two specific questions I wanted to answer initially using play-by-play data:

* How accurately can we predict total points per game?
* Can we reliably predict whether or not the home team wins?

The first big question was whether I wanted to approach the prediction from the pre-game perspective or the within-game perspective. The pre-game perspective answers the question “given all the games and plays to-date this season, which team do we predict will win this game and what will the total score be?” This is the purely ex-ante perspective akin to predicting the future based on the past. The post-game perspective answers the question “given all the games and plays over all seasons, including this game, which team do we predict will win this game and what will the total score be. The former has a limited sample size but a more valuable (and challenging) question to answer, while the latter has a many-fold larger sample size and a simpler (though less meaningful) question. I decided to go with the latter because I had no prior experience using Python for data analysis and thought it best to focus on basic ‘learning to program’ issues rather than complex analytical problems; though the ‘pre-game’ perspective is something to look at later on.

Finding the most reliable source for play-by-play data was the first big challenge. The NFL and ESPN have no APIs for box scores or other data, so page-scraping was the only option if I wanted to use those sources. However, the one source actually looking advanced NFL metrics makes the play-by-play data they use from 2002-2012 publicly available (Burke, 2010). This is my primary data source, containing:

* Basic down, distance, time, yard line, home & away, score, date
* Play description stating the play type, yards gained/lost, turnovers

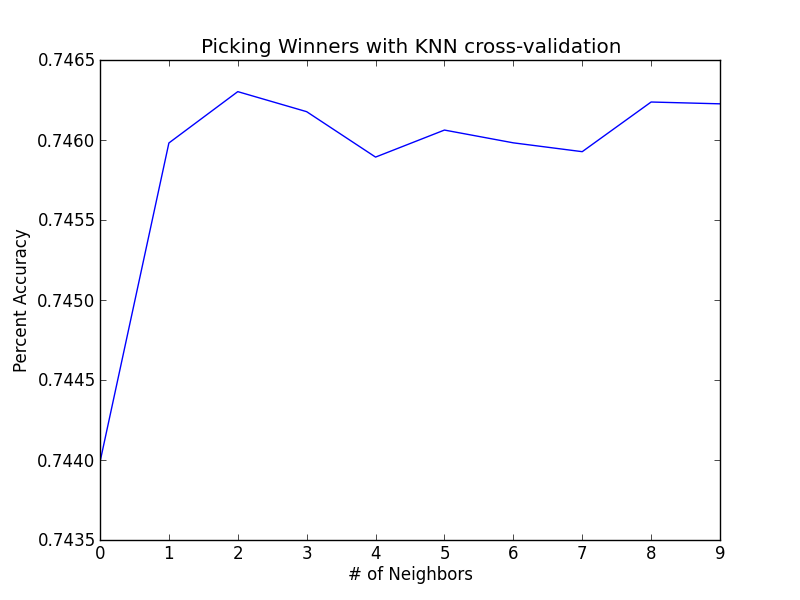
Additionally this data allows us to calculate other variables like yards gained, time of possession and penalty yards. These are potentially important predictors of game outcome as the more yards a team gains or more time they have the ball the more likely they are to reach the end zone, and the opposite effect for penalty yards. Data on the location and type of each team’s stadium was gathered to account for the effect of travel and domes on the game (Wikipedia).

The distinct questions here necessitate two different analytical methods. To predict the winner I chose the K-Nearest Neighbors classification with n-fold cross-validation, since this is a classification style problem. To predict points I used the linear regression and prediction package within python’s ‘sklearn’ package. I’m also using a linear model (as opposed to a Ridge or Lasso model) because the concern for overfitting is low with the large sample size (2684 games) and moderate number of predictors (23). The linearity of the data is backed up below.

**Setup and Analysis**

The first (and by far the largest) task was to inspect and clean the data, some of which was randomly blank (like play start times) and other times systematically blank or incorrect. In the case of games that went into overtime, there were many that were missing plays altogether or had blank play descriptions; on top of this handling time in overtime games (since there’s no definite period, it’s sudden death) presented another challenge. Since the data couldn’t be trusted as unbiased I excluded all games that went into overtime.

The first of the two analytical tasks I tackled was the win-loss classifier. I chose an n-fold cross-validation approach with the K-nearest neighbors method because of the diverse and deep dataset. With the seed set, I chose a 70/30 split (an 80/20 split revealed no further gains) and randomly chose 70% of the data as a training set and 30% as a test set, running it 100 times and averaging the results. The predictive accuracy converges on 74.6%, which, even though there’s no reference point, seems good. What was most interesting was when I varied the number of neighbors to analyze; moving from the default of 5 neighbors to a range of neighbors from 1 to 10 resulted in no significant changes in accuracy (see graphic below, note the scale on the left). It seems probable the breadth and depth of the data minimizes the value of additional neighbors being used.



The second task was to estimate total points per game, using linear regression with backwards elimination. This requires the ‘kitchen sink’ approach at first, tossing all variables in and seeing what’s worthwhile. For this model my overall measure of accuracy is the absolute value of the percent error, represented below:

*i* /Pi)

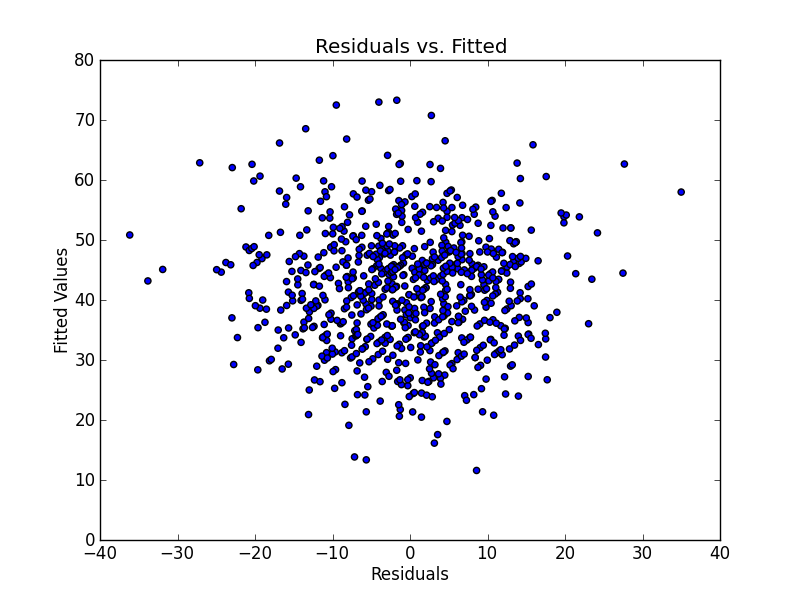
Where:

n = total predicted observations

ε = the predictive error for observation

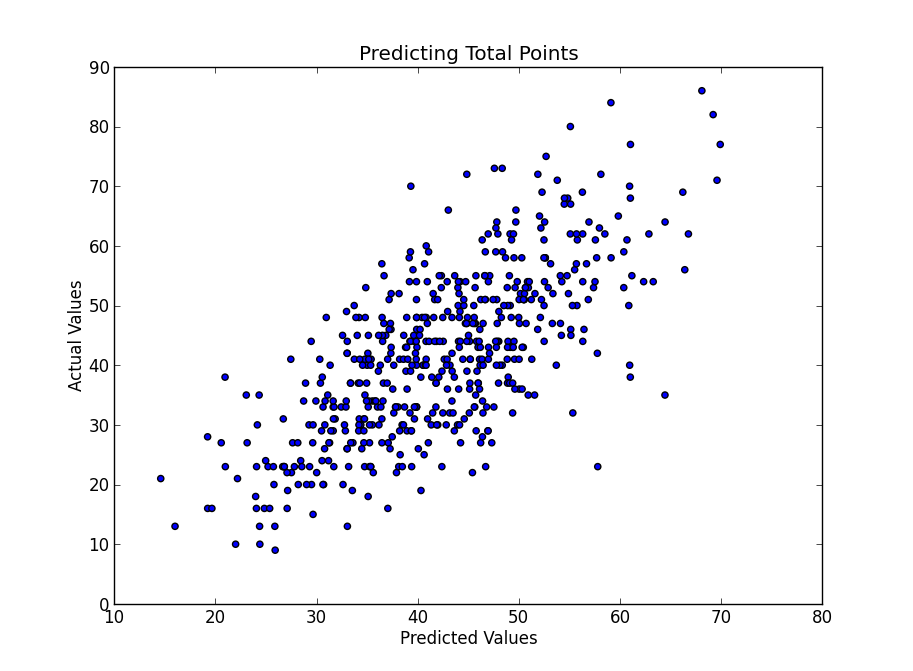
P = actual points for observation

The ‘kitchen sink’ approach yielded a first-pass absolute percent error of 37.7%, which doesn’t seem great but doesn’t seem terrible either. One thing we want to do here is plot the residuals and the fitted values to make sure we can’t detect any bias. We want it to look random, with no detectable pattern.



This is exactly what we’re looking for; confirming linear regression is a good format to work with. Looking at the coefficients and standard errors, ‘season’ looks like the best candidate to drop first. Season is just a factor variable that attempts to account for season-by-season changes in total points. After removal, the error rate doesn’t change; even though the error rate didn’t improve, that it didn’t get change at all supports its removal. The next worst variable is ‘playoffs’, which is a binary variable indicating whether or not the game is a playoff game. It is perfectly correlated with month, as the last month is always the playoffs. After removal this brings us down to 37.6%; no real improvement, but again because it didn’t go up, it supports the hypothesis that it shouldn’t be in the model. ‘Month’, which to is somewhat of a proxy for temperature, but also tries to account for how teams play as the season progresses, and during the playoffs, still looks insignificant. After removal, we’re at the same level of accuracy, 37.6%. The final variable that looks insignificant is ‘Sunday’, which is a binary variable indicating whether or not the game is played on a Sunday. Games not played on a Sunday (Monday and Thursday) are games in which the teams playing are in the spotlight since they’re on national television, and would arguably want to play even better than they usually do. However after removing it, we see no improvement.

Now that we’ve got our final model, we want to create a scatterplot as an extra step to confirm the use of a linear model.



This is what we want to see, a tight cluster in the form of a line through the middle.

**Conclusion and future research**

While I didn’t expect to be able to predict points well, picking 75% of the winners based on a plethora of in-game data seems low upon further reflection, and highlights how random a sport like football really is. One important factor missing from this exercise was weather data. The impact of temperature (see the effects of ‘month’) and precipitation cannot be understated. It’s much harder to complete passes, gain foot traction or maintain ball control in inclement weather, which depresses scoring output. Quantifying temperature, wind and the amount and type of precipitation is a logical next step. Weather underground has an API that can give historical data for a given zipcode and date, so learning to build that is a next step and will certainly improve accuracy.

The other avenue for future exploration is pre-game prediction: predicting the winner and total points based on our to-date knowledge for that season. This has obvious applications for betting markets, in-game commentary and fantasy football. One would utilize similar variables, but the analysis would be a team-centric approach rather than a game-centric approach as seen here, and you would obviously be unable to include the kind of in-game data (yards gained, penalties) that are so valuable with this kind of model.