Project Proposal

Chris Fenton

DATA 698

**Context / Problem Statement**

One of the challenging things about doing data analysis of NFL Football is that since the sport is highly situational, it makes many of the basic game data hard to value without proper context. Certainly, a scoring touchdown is generally going to be a positive data point for a team, as it scores points for that team, and the object of the game is to score more than other team.

However, consider a player running for 4 yards (one of the more common plays of a football game). Is this considered a positive or negative play for that player’s team? The answer completely depends on the context. Is the player’s team facing a 3rd and 1, where four yards will bring a new set of downs? Than very likely it is a positive play. However, if the players team is facing a 3rd and 20, they will likely have to punt, thus giving the ball away, and it is a negative play. So the data that we have that a player ran for four yards is impossible to assign proper value to without additional, contextual data.

Despite this truth, the way the NFL community has typically analyzed game statistics ignores context, out of either a lack of availability, ignorance, or both. After all, it is much easier to calculate the total amount of yards a player has run for, than to go play by play and manually assign each play a value of “positive” or “negative”. And generally speaking, gaining yards is an indicator of success, since it states that the ball was moved toward the opponent’s goal, and scoring is the object of the game. But if we stop there, much of the true information about what happened is lost.

**Literature Review/Research Conducted**

In recent years solid progress has been made addressing this via a variety of efforts that attempt to encapsulate critical “game state” information that can be used to contextual game data. One of the first was Defense-adjusted Value Over Average (DVOA) by Football Outsiders (<http://www.footballoutsiders.com/info/methods)> , which manually encoded each play according to a proprietary scoring scale, and summed the results by player, game, or season.

A more recent attempt has been to incorporate the idea of Win Probability (WP), and its close companion, Win Probability Added (WPA). The idea of Win Probability goes back to mid 20th century baseball, but was widely introduced to NFL analysis by Brian Burke at his former site Advanced Football Analytics (<http://www.advancedfootballanalytics.com/index.php/home/stats/stats-explained/win-probability-and-wpa>). Since then, a variety of other approaches have emerged, summarized by Michael Lopez at his personal blog (<https://statsbylopez.com/2017/03/08/all-win-probability-models-are-wrong-some-are-useful/)>.

Although the models surveyed by Lopez include a variety of implementations with regard to both data and model selection, most follow a similar approach. A sample of recent NFL plays are collected (typically no older than from the 2000 season, as that seems to be an inflection point where NFL data became widely available). These data sets will include play by play game logs that include game state data encoded, such as (but not limited to):

* Home Score
* Away Score
* Score Differential
* Total Points Scored
* Down
* Yards to go (for first Down)
* Time Remaining
* Las Vegas Point Spread
* Timeouts Remaining

Once play data is selected and encoded, a model was fit on the data. Varying levels of transparency have been provided, but among the approaches were Random Forest (Lock and Nettleton), ensemble methods (Burke), logistic regression, and others. The common denominator amongst all methods were they were classifiers. This is natural, since within this framework the outcome variable of interested is whether a team won or lost (ties are rare and were ignored).

**Theory**

My goal for this analysis was to draw on the aforementioned research to fit a well performing Win Probability estimator model, fit that model to a series of plays, and use the difference in Win Probability for consecutive plays to derive Win Probability Added.

From there, my hypothesis was that a positive aggregate WPA score over a given subset of plays (game/player/season) would have a stronger relationship with wins than traditional statistics.

**Data**

The data I used for my analysis comes from Andrew Gallant’s nfldb (<https://github.com/BurntSushi/nfldb)>. Nfldb scrapes NFL.com’s (the official league website) JSON feed, parses, and stores the data in a relational Postgresql database.

A condensed view of the Postgresql data model is shown in the below nfldb Entity-Relationship Diagram (Figure 1):

*Figure 1*

The key tables for the purposes of this analysis are the game, play, and play\_player tables. Game contained contest information such as which year the game took place, who the participants were, and the score of the participants (particularly important for deriving the outcome independent variable). Play included each situation, such as down and distance and time remaining. Play\_player included player performance on each play. This was used to derive the score for each play, something that was not available “out of the box” from the database, thus determining if any player contributed to a score was a way of determining whether a score occurred.

My variable set used for model fitting was as follows:

**Offense Win (Outcome):** Indicated whether the team that possesses the ball (the offense) wound up winning or losing the game, with 1 for a win and 0 for a loss.

**Offense Score:** The point total for the possessing/offensive team.

**Defense Score:** The point total for the non-possessing/defensive team.

**Down**: 1, 2, 3, 4, or 0 for non down plays (such as kickoffs). If the requisite number of yards aren’t gained by 4th down, ball possession changes hands.

**Yards to go:** For down plays, how many yards are needed to either gain a first down or score.

**Game Seconds:** How many seconds have elapsed from the game clock. Each non Overtime NFL game has 3,600 game seconds, thus 0 seconds would indicate the beginning of the game, and 3,600 would indicate the end of the game.

**Field Position**: Where the ball was on the 100 yard long football field. -50 indicates a team on its own goal line, 0 indicates the ball at the 50 yard line (mid field), and 50 indicates the ball close to the opponents goal, thus the possessing team is close to a score.

Initially, instead of Offense and Defense Score, I used Home and Away Score. These were used in conjunction with a third variable, Offensive Possession, which served as an interpreter of sorts to tell the model whether Home or Away score pertained to the offensive team, since the outcome variable, Offensive Win, was measured in terms of the offensive team. I changed to Offense and Defense score to eliminate this extraneous variable and simplify the model.

While some of the mentioned variables were readily available in the nfldb database (such as time), some needed to be derived (such as score). Further, these variable did not all live in the same table, and instead needed to be joined and in some cases aggregated from several tables.

Along with the database, nfldb includes a python package that serves as an interface to the data model, but I decided it would be easier to query the database directly via SQL. My process for getting the data from the format in figure 1 to the previously discussed variable set was a combination of SQL queries in R and dplyr package data manipulations. I wrote a series of functions to do these tasks, and a script that called these functions to build the R dataframe used for training.