NFL Win Probability Modeling*

Shawn Leahy
Carnegie Mellon University
Pittsburgh, PA
sleahy@andrew.cmu.edu

David Brennan
Carnegie Mellon University
Pittsburgh, PA
dsbrenna@andrew.cmu.edu

ABSTRACT

Sports analytics have been generating increasingly more attention amongst organizations in recent years. From 'moneyball' in baseball to tracking the movements of players around the field, analytics are being applied more and more to any aspect that can yield a competitive advantage. Building on this trend, we use the machine learning problem solving methodology from 95828-A to create more accurate probability models used to evaluate how each play influences the outcome of the game and develop metrics that can be used to accurately call the results of a game prior to time running out.

ACM Reference Format:
Shawn Leahy and David Brennan. . NFL Win Probability Modeling. In *Proceedings of Heinz College (MLPS 2018).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

A win probability is the probability that one team or group will win a game or event based on the conditions (score, time, etc.) at a particular point of time. Players, coaches, and fans in the realm of sports undoubtedly perform similar calculations implicitly as the game progresses. For football, this win probability can be used to evaluate the probability of winning the game prior to each play. Additionally, this type of model could be used to evaluate play-calling and certain other coaching decisions in real time. Furthermore, they can provide context to fans by improving the viewing experience throughout the game.

The concept of modeling probabilities as an event progresses is not new or unique to sporting events[13]. Win probabilities have been used for political forecasting such as on FiveThirtyEight[12] or the Princeton Election Consortium[15]. It has been used for a range of sporting events from College Basketball, Major League Baseball, and Football amongst others[3][11].

As is often the case of things, win probability models only generate discussion when they have been outrageously wrong. In the 2016 United States Presidential Election, many election forecasters had democrat Hillary Clinton with a 95+% chance of winning the election. Nate Silver of FiveThirtyEight had perhaps the most accurate of the wildly known election models, but even his had Clinton with a 66% chance of victory[12]. In the sports world, the Cleveland Cavaliers came back from a 3-1 game deficit to beat the record-setting Golden State Warriors in the 2016 NBA Finals. Also in 2016, the Chicago Cubs rallied back from an eerily similar 3-1 game deficit to beat the Cleveland Indians in the MLB World Series. These phenomenon happening in such a short time span gives rise to an interest of exploration into the world of win probability modeling[14].

The primary motivation for this paper comes from the events of Super Bowl 51 between the New England Patriots and the Atlanta Falcons. With two minutes left in the third quarter, the Patriots were trailing 28-3. Most probability models had the win probability of the Patriots somewhere around 0.1% or 1 in 1,000[6]. ESPN counted over 20 different points in the game where New England's win probability was 1% or less[1]. Despite the models, New England pulled off the comeback to win the game. These results do not mean win probability models are flawed, rather these high profile events are edge cases for model prediction and a motivation to work in this domain.

One of the most well known Football win probability models comes from Pro Football Reference (PFR), one of the leading sports data repositories in the world. PFR's win probability model uses the assumption that the score differential throughout the game can be approximated by a normal distribution with a mean equal to a combination of the Vegas line for the game as well as features occurring on each play. The standard deviation changes as the game progresses. PFR also incorporates a second model for the last 5 minutes of each half to account for changes in strategy that occur in those times[10].

The literature of this topic also includes experiments by Dennis Lock and Dan Nettleton into the effectiveness of various algorithms for win probability prediction where they show the effectiveness of the Random Forest algorithm[7]. We follow this approach for part of the analysis in this paper. Other algorithms have shown to be effective probability calculation tools such as Support Vector Machines and Logistic Regression[8]. We employ Logistic Regression in the second stage of our analysis, though in a slightly different capacity from the established literature.

1.1 Problem Definition

A useful indicator of the accuracy or efficacy of a win probability model would be if the percentage of the in-game plays where one team had a high probability of winning, but did not win was minimal. For example, the New England Patriots were given about a 0.1% probability to win Super Bowl 51 at the end of the third quarter by several well known models[10]. This number tells us that for every one thousand times that this situation is repeated with a team in the same position, there will be one occurrence where the team with the significantly lower probability will win. As noted above, for election forecasting, many forecasters had Donald Trump losing in 100 or more simulations before a win[15]. There has not been literature done to our knowledge of simulating a series of games (such as a 7 game series of basketball, hockey, or baseball) but based on anecdotal occurrences, a 3-1 advantage is formidable. This possibility of a comeback can happen at any point, so it could be that the

Table 1: Example Duplicated Variables

Variable Name	Description
Time	Minutes:Seconds representation
Time in Seconds	Seconds elapsed in the game total 0 to 3,600
Yard Line	On-field value of yard line play began on
Yard Line from 0	Yard distance from defending end zone
Absolute Score Difference	Absolute value of score differential
Score Difference	Actual Score differential of possessing team

Table 2: Filtered Data Results

Description	Counts
Total Number of plays ran:	267,788
Total Number of downed plays ran:	267,788
Number of Touchdowns:	10,268
Number of Safeties:	112
Number of Interceptions:	3,850
Number of Penalties:	3,950
Number of Field Goals:	7,767

first time we ever observe this game situation (i.e. a team trailing with 25 points with less than 17 minutes to play)[4] the unlikely event occurs. Our model seeks to make reliable estimates of win probability, while also providing a threshold where we would be able to consistently make an accurate call of the result of the game prior to the game actually ending. Machine learning will allow us to model the in-game probabilities, but will also allow us to validate a series of thresholds for result prediction based on those probabilities. These thresholds will be sampled randomly, but uniformly, and will have set weights. Then we can check how accurate our model performs in a variety of situations. However, we do not seek to directly model these unlikely events, despite all the discussion. These events are a strong motivation, but represent outliers and attempting to fit these cases into a model would be akin to fitting the noise in your data.

2 DATA PREPARATION

2.1 Data Sources

Our data contains every play from every regular season game from 2009 to 2016 obtained from Kaggle. The data was compiled using the R package NFLscrapR created by statistical researchers at Carnegie Mellon University[5]. The data is scraped directly from NFL.com. The data is approximately 239.3MB containing 102 variables and 362,447 observations. Each observation in this data is a play or event that occurred during a game. This totals to 2,048 games.

2.2 Data Integration and Preparation

Upon initial investigation of the data, the first step was to go variable by variable and remove those we knew would not be relevant to our modeling efforts. Removed variables at this stage include unique player information relevant to the play and date of the game. Our modeling approach does not care what players were involved in a play, only the outcome of the play in regards to score, field position etc. There were also several variables that were in some ways duplicated or transformations of other variables. These are shown in Table 1

Many of these variables were removed. *Time in Seconds* is preferred to *Time* because then each play for each game will have a unique value, whereas *Time* will repeat every quarter. *Yard Line from 0* gives an absolute reference to a play's starting position on the field as compared to *Yard Line* which will result in an ambiguity as to whether the play began on the offenses of defenses side of the field.

We also remove variables that involve details of the play that combine together to form the total result of the play. Instead of including the yards a pass traveled in the air and the yards after the catch a receiver ran for, we only include the total net yards of the play.

After the initial cleaning steps performed on the data, we append the result of the game to each play. For each individual game, we identify the winning team and the losing team at the end of the game. We then append an additional variable, *result* ,to each play that identifies 'W' if that team running the play won the game, 'L' if the team running the play lost the game, else we append 'T' for the game ended as a tie (discussed later). These will act as our true labels at the classification step of the report.

The next steps in the data cleaning involve including only plays with the offense and defense on the field. This means removing special teams plays, extra points/two point conversions, and 'plays' without a down that include values such as 'Two Minute Warning ', 'Quarter End', 'Game End' etc. We also exclude overtime from our analysis, though this could be added in later on with some extra steps and no loss of accuracy. We convert field goals into a series of dummy variables including: 'Good', 'Blocked', 'No Good'. Similarly, all variables indicating the results of a play i.e. touchdown, safety, interception etc. are transformed into dummy variables.

Once this preprocessing is complete, we remove for now (but keep so as to be available) variables including the unique game ID, the home team, and the away team. This will be used after the models are computed in order to look at how well our models perform on individual games and gives context and interpretability to our analysis. Our labels are the win/loss indicators for each play that shows the result of the game for the team running a given play.

It is on this pre-processed data that a statistical investigation of the structure is conducted. Some results are shown in Table 2. Our results confirm the data behaves as expected within reasonable limits. the highest volume of plays occur at the beginning of each quarter/half and at the very end of the game (Figure 1). Most of the plays are classified as run/pass plays and a reasonable number of other plays are classified otherwise. We find that there are less 'half-end', 'quarter-end, etc. 'plays' than there are number of games/quarters. Upon investigation this is due to these 'plays' only occurring when the clock runs out and no play is in progress. If a play is in progress as a half ends then this designation does not appear (Figure 3). Other results include most plays being run in the middle of the field, 40-60 yards from the offenses end zone, and more plays are run on first down than second, more on second than

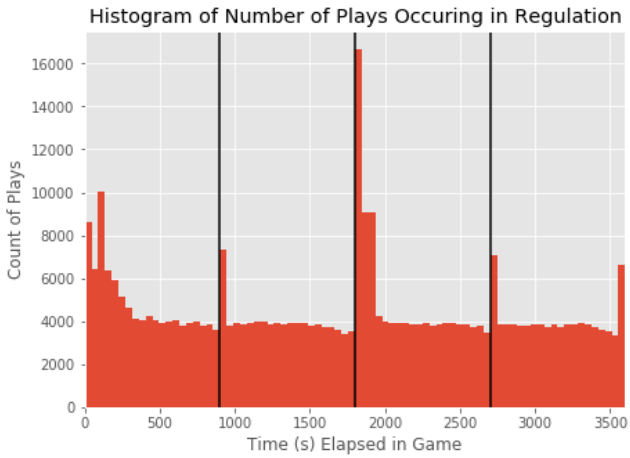


Figure 1: Number of Plays Occurring Per Second in Regulation

third etc.(Figure 4). Finally, most score differentials fall in the 3-14 point range as we would expect in a football game (Figure 2).

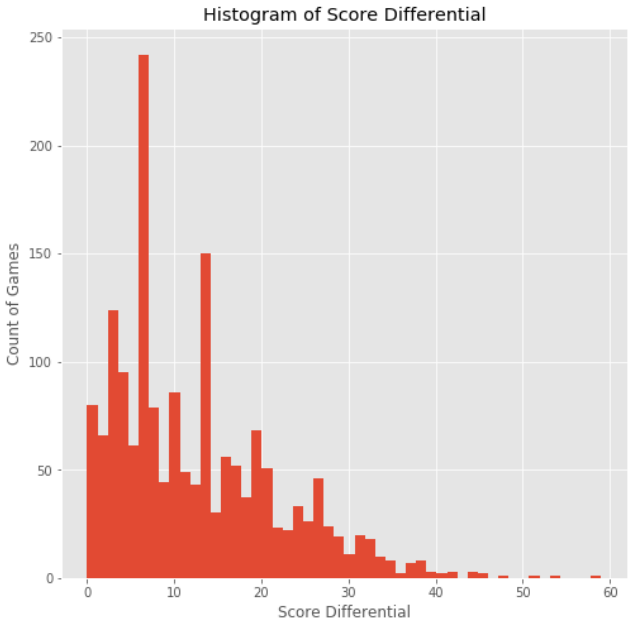


Figure 2: Number of Plays Occurring with a given Score Differential

3 METHODOLOGY

3.1 Model Approach

The first step for conducting this analysis, and arguably the most important, is obtaining probability estimates for each play indicating how the result of said play affects the possessing teams probability of winning the game. To get this result, feature selection needs to

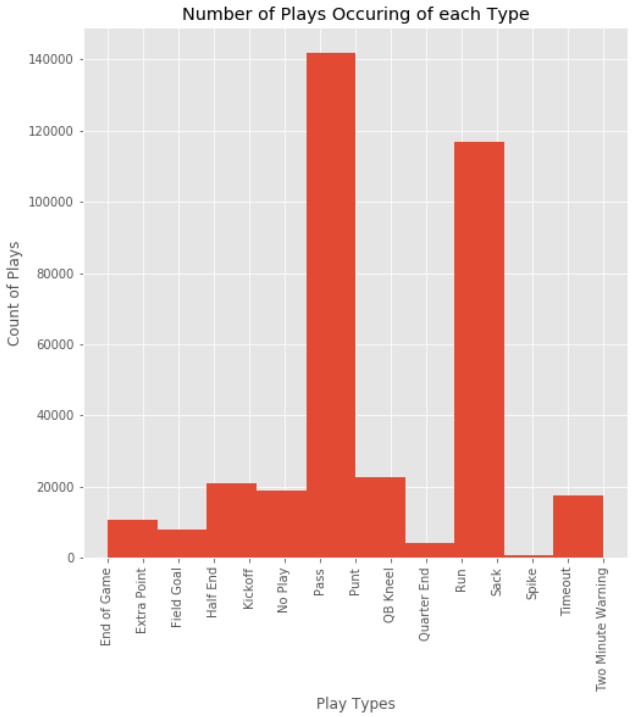


Figure 3: Types of Plays Ocurring in Data

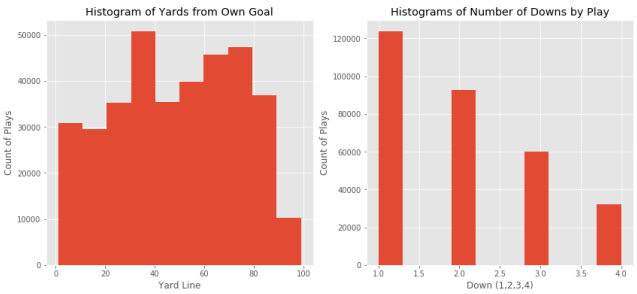


Figure 4: Position on Field Plays Occur at and Plays by Down

be conducted to obtain features that influence the outcome of the game. Some of the most important variables to include:

- *Score Differential*: We define score differential by the team possessing the ball. This ensures consistency in what the values mean. Each play of our data will have a point difference, a positive value indicates the offense is winning, a negative value means the offense is losing, and the teams are tied if it is zero.[4]
- *Down and Distance to go*: Identify what down it is and the number of yards to go to reach the next first down.[4]
- *Field position*: We use this feature from the offensive perspective. We will turn field position into a value measuring the distance from the offense's end zone. The offense on its own 25 is coded as 25 yard line and the same position on

the defenses side of the field would be the 75 yard line. This reduces ambiguity and duplicity in the data.[4]

- *Time*: Since minutes and seconds are repeated for each quarter, we cannot use them due to ambiguity (or needing multiple variables to gauge time in game). Thus we will start the game at 0 seconds and code this until the end of the 4th quarter at 3600 seconds.[4]
- *Result*: To conduct a supervised machine learning approach, we need to know what team wins the game and what team loses the game for each play. This will be coded as 'W' and 'L'.[4]

Other features included in the model include scoring indicators (touchdowns, field goals, safeties), net yards gained on a play, penalty yards, interception thrown, and timeouts per team indicators. These may be adjusted incrementally depending on results from our models.

There were several further challenges encountered, beyond those noted above prior to the ability to implement a workable model. Many variables were encoded categorically, so we turned them into a series of dummy variables using one-hot-encoding. It was also challenging to encode the result of each game for each team onto each play in the data. To do this we needed to construct a dictionary with keys of the game ID and values including the name of the winning team. A similar process was conducted for the losing teams in the data. We then need to iterate through the entire dataset, taking the game ID of each play and checking if the possessing team is in which dictionary. There are also many features that have *N/A* for most plays as they occur only rarely in a game (i.e. blocked field goals) the models cannot take in features lacking values so for each feature we needed to check how many and why they have *N/A* values. This was performed incrementally over the features. The *Play-Type* variable was instrumental in identifying why certain plays lack variables and how to filter and arrange the data to control for this.

As noted above, there are variables that are not useful for modeling purposes, but we need to keep in the data up to the modeling stage because they can be useful later in the analysis. Unique game identifiers, team names, and season all are kept. The data also has a proprietary win probability algorithm from Pro Football Reference included that we will compare against. We will keep these values outside the models, but can use them to compare how our results differ from established methods.

Feature engineering to transform variables could be useful in capturing features not present in the default data. One such transformation involves the fact that time in game effects win probability more and more the closer we get to the end. A team down by 3 points in the first quarter is not the same as being down 3 points with a minute left in the 4th quarter. It would be useful to have a variable that stays relatively constant but starts becoming more and more significant as the number of seconds remaining increases[4]. For this feature, we'll use:

$$\frac{ScoreDif}{sqrt(SecondsRemaining + .01)} \tag{1}$$

Next we need to justify removing post season games from our analysis. The reasoning behind this is we assume all regular season games are generated from the same process. In football terms, all decision made by players and coaches are operate under the same rules and similar decision making. Now this may not hold up for the final few games of the season in some cases where team's are in a win or their season ends scenario, but for the most part this assumption holds up reasonably well. The playoffs are different. Since one loss will end your season in each and every game, different strategies may be utilized by teams in desperation. Our assumption that all games are generated from a similar process is violated in this case[4].

We call generating these probabilities for each play the first stage of our analysis. The next stage is that in order to make a claim that we have a *good* estimation of win probability model, we will need to predict games prior to the end. In order to classify the outcome of a game, a prediction is needed well before the outcome of the game is decided. The ideal scenario would be analyzing several hundred games of data and working backwards to see if there is a time and a set of win probabilities from the game that can predict the outcome.

We will construct a function $f(w_{probability}, t_o, w_{weights})$ that takes in a set of win probabilities sampled from the game, an offset time from the end of the game, and a set of weights for the probabilities and outputs a prediction of 1/0 for win/lose at some offset time before the end of the game (i.e. 5 minutes, 7 minutes).

3.2 Models and Algorithms Stage 1

We utilize all plays from 2009-2015 as our training set, and use the 2016 season data as our testing set. We employ a Random Forest model in order to predict win probabilities for each play.

Random Forests as an algorithm are useful in this situation because they don't over fit as easily as many other methods and allow for non-linear interactions between features[7]. Ensemble models are very efficient at reducing variance, and with data where there are many observations that are alike in many ways, but also different in many ways, the ability to generalize to new data is incredibly important. It also allows us to estimate how important certain attributes are for predicting win probability. If a feature in our model is removed and our accuracy for labeling wins and losses does not decrease, that feature might not be a significant indicator to our model. We use the Random Forest algorithm in order to create win probabilities for each play and perform cross validation in order to identify optimal hyper-parameters.

Figure 5 shows the change in win probabilities for a single game. We randomly choose a game from our testing set (the 2016 season), and we end up with the Seattle Seahawks at the San Francisco 49ers. The game resulted in Seattle winning 25-23. The blue curve indicates the win probability from Pro Football Reference (PFR)[10], while the red line is our predicted model. We observe our predicted model closely tracks the industry standard, with a slight increase in volatility at points (this could be due to some modeling differences noted in the introduction). These differences are difficult to explore without more information on the PFR model, but we can try and track the magnitude of these differences. We take the mean absolute error between the two models and track this over all games in Figure 6. Due to the fact that the ith play in one game may not perfectly correlate with the ith play in a different game, we cannot build an

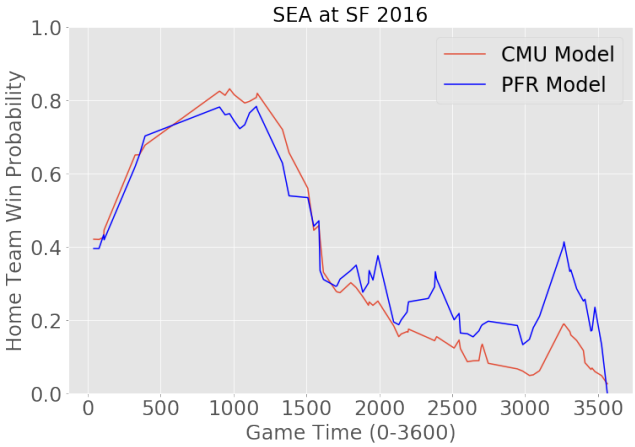


Figure 5: Win probability for a Random Forest model versus Pro Football Reference baseline model

aggregate error function across all games (in fact the number of plays in different games is far from constant). What we instead set out to do is measure the absolute difference between our model and PFR's model on a play by play basis, obtain a game average error and plot across all games in our test dataset. We observe the differences range from a 2% difference up to a 10% difference with a mean around 4%.

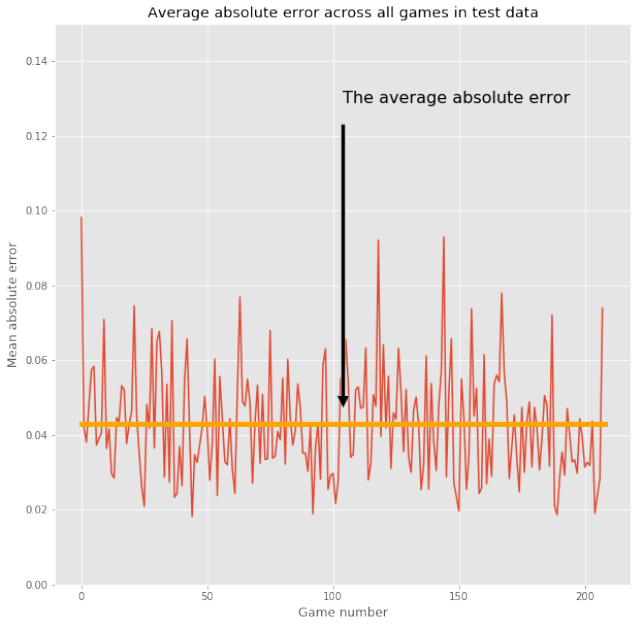


Figure 6: Difference in Win Probability from PFR to CMU Models Over All Games

3.3 Models and Algorithms Stage 2

Predicting the outcome of a game merely using win probability is flawed as we simply turn the problem into a threshold cutoff based approach which we can clearly convince ourself is not the case for making good predictions (as is evidenced by examples cited in the introduction).

A possible prediction method would be to take a weighted average $\beta_1, \beta_2 ... \beta_n$ of the win probability sampled at various times $t_1, t_2 ... t_n$ (either deterministically or stochastically). That is to say we could model the probability of a home team winning a game by:

$$\beta_1 * w_{att_1} + \beta_2 * w_{att_2} + ... + \beta_n * w_{att_n} = \Phi \tag{2}$$

If $\Phi > .5$ we would predict a home team winning, otherwise the away team would be predicted to win. We would also need to impose the constraint of:

$$\sum_{i=1}^{n} w_i = 1 \tag{3}$$

in order to ensure the probability $w_{home\ win}$ is $0 \leq w_{home\ win} \leq 1$

A simple (but not quite elegant predictor) could be taking the win probability of the home team sampled at halfway point of each quarter, or taken at $t = 7.5/min, 22.5/min, 37.5/min, 52.5/min$. Knowing that this doesn't embed any of our existing knowledge of time playing a critical role in determining win probability, we would choose to not naively have $w_1 = w_2 = w_3 = w_4$ or $\forall w_i = .25, for i \in 1, 2, 3, 4$ as the weighting scheme. This weighting scheme would show equal preference towards having a high win probability at any point in the game which is simply not correct. Expanding upon this observation, the reader might have picked up on the fact that it would benefit the model to have a strictly monotonic arrangement of weights, that is to say this scheme would exhibit the following behavior:

$$w_1 < w_2 < ... w_{n-1} < w_n \tag{4}$$

This gives more power towards having a higher win probability towards the conclusion of the game which conforms with the reality of win probability (it's better after all to be winning closer to the end of the game as opposed to the beginning if you had to choose merely one!). The above is merely one possible path to take when it comes to weighting the result. Using this method, there area large number of "weightings" and times at which to sample the win probability exist and it would be infeasible to search over all of them for all games.

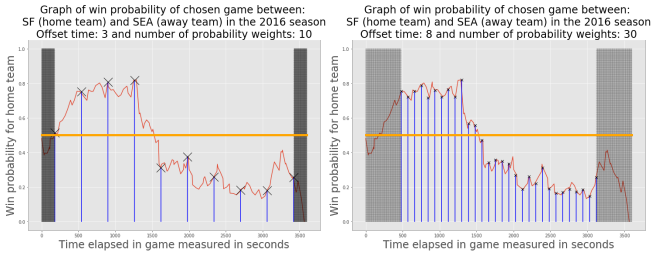
Instead of manually setting the weights in a the naive manner specified about, it would be advantageous to use another machine learning model. Logistic regression is an effective classification methodology that we use in the second stage of our methodology instead of the above naive approach and would provide a more optimized set of weights. The logistic regression formulation calculates the probability of the home team winning given a vector of win probabilities sampled from the game.

$$Pr(R = 1|\mathbf{x}) = \frac{exp(\mathbf{w}^T * \mathbf{x})}{1 + exp(\mathbf{w}^T * \mathbf{x})} \tag{5}$$

R is the dependent result variable of our model representing if the home team wins or loses, $\mathbf{x}$ is a vector of probabilities sampled from various points in a game, while the coefficient vector $\mathbf{w}$ includes the weights for each probability [8]. We again create two hyper-parameters, an offset time and a number of probabilities to use as

input. For each combination of hyper-parameters we iterate through each game and sample the number of probabilities while excluding the time specified by the offset time from the end and beginning of the game. The weight sampling is initially done at the beginning and end points specified by the offset time then uniformly between those points. We want to make a prediction prior to the end of the game so we exclude a certain time from the end of the game. We remove the beginning of the game as we want to remove any possible outside influence affecting the probabilities (i.e. a model using Vegas win odds to set a prior probability before the game begins). Then the rest of the probabilities are sampled evenly from the remaining time in the game. Figure 7 shows examples of this methodology for a single game using two different sets of hyper-parameter values. We observe the left plot having a low offset time (close to the beginning and end of the game) and only a few sampled points, while the right plot has a high offset time (further from the end of the game) and many sampled points.

Once we iterate through all games for a set of hyper-parameters we will essentially have created a new dataset with one row for each game and one feature for each sampled probability. Using the outcome of the game as labels, we are able to fit a Logistic Regression model to the data. Similar to the Random Forest model from stage 1, we split the data into a training and testing set. Fitting on the training set and predicting on the testing set, we can evaluate how well our predictions perform relative to the true labels. This process can be done multiple times for each desired set of hyper-parameters.



(a) Low Offset Time, Low Samples **(b) High Offset Time, High Samples**

Figure 7: Comparison of Stage Two Hyper-Parameters

3.4 Model Optimization

Model optimization is possible with both the Random Forest models and the logistic regression models by finding optimal hyper-parameters. For Random Forests we perform a grid-search 10-fold cross validation to find an optimal depth which happens to coincide with a depth equal to ten.

For Logistic Regression we perform a similar cross validation mechanism to determine the value of the regularization parameter to reduce over fitting. Using this process we are able to find a locally optimal parameter setting for model prediction.

We are hesitant to optimize the hyper-parameters for offset times and number of weights in the same manner. An attempt to perform the same cross-validation procedure to maximize accuracy would result in the offset times being set almost to (or at) zero

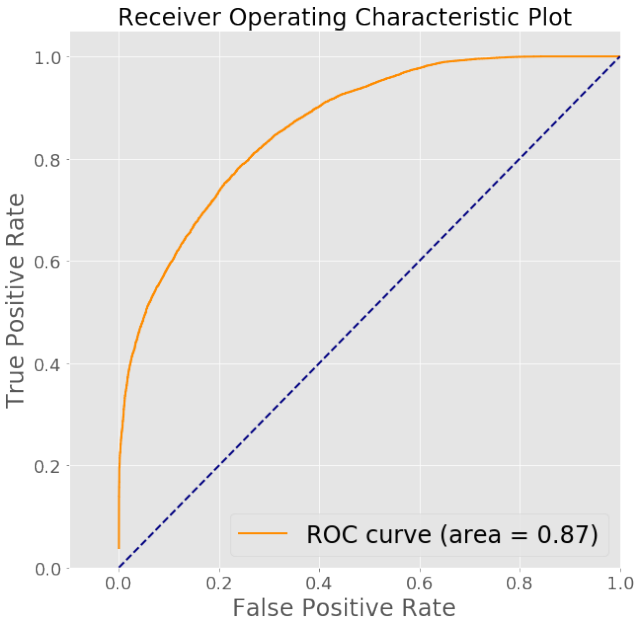


Figure 8: ROC and AUC for Random Forest Model

and our results would be meaningless. Also, for this mechanism we are not necessarily looking for the most accurate set of hyper-parameters, rather we are looking for an interesting set of hyper-parameters. Thus we perform a grid search over a set number of possible variables to illustrate how the methodology performs under a variety of situations.

4 EVALUATION

4.1 Stage 1

Since the win probability from the PFR uses features that may or may not be contained within our dataset we do not know what features the model is predicting from[10]. We also realize and acknowledge that these are merely estimates, the actual win probability of an NFL team is close to impossible to ascertain for certain.

Figure 8 illustrates an ROC curve showing how the predicted probabilities map to the actual labels (0 is a loss, 1 is a win). We can see that the model results in an AUC value of 0.87. The accuracy prediction for the model was 77%, thus the combination between these two metrics lends itself to show the model is performing adequately and the data-set is not skewed. Combine this with the similarity between our model and the PFR model, and there exists reason to believe this model is performing as desired.

4.2 Stage 2

The sample space for possible permutations of offset times and number of weights is large. Figure 9 shows a heat map of accuracy scores over a grid of hyper-parameters by utilizing the logistic regression methodology specified above. We use 1,486 games as our training set and 208 games as our testing set. As we set the offset time to be further away from the end of the game (descending the y-axis), our predictions become less and less accurate. What

is interesting from the figure is that the number of probabilities sampled (x-axis) shows no discernible pattern as to how accurate an outcome can be predicted.

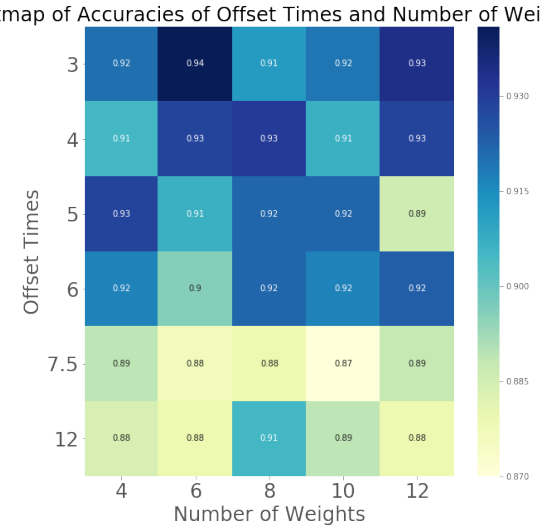


Figure 9: Game Prediction Accuracies for Different Offset Times and Weight Numbers

We can create a similar type of measurement visualization using different metrics. Figure 10 shows the average precision scores for each combination of hyper-parameters - offset time and number of sampled probabilities. Where the accuracy metric measured how often did we get the winning and losing teams correct in a game, using the precision score will be able to tell us how often we labeled a team as the winning team that ended up losing the game. A higher precision score is indicative of more of your positive values (in this case winning) being true positives rather than false positives[9]. From the scores in Figure 10, this set of parameters has precision scores ranging from 84% to 95%.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{6}$$

The F1-Score is an accuracy metric that takes the form of the harmonic mean between the precision and recall. Precision and recall for the F1-Score are equally weighted, as we do not assign any higher or lower penalty for false positives (this would lead to using an F2-Score or other variant)[9]. Figure 11 shows the grid of hyper-parameters and their F1-Score. Our evaluation metrics are fairly similar, with the F1-Score ranging from 89% to 96%.

$$F_1 = \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

4.3 How does the Model Improve Problem

It is impossible to create model that perfectly predicts the outcome of every game. As is stated in the introduction, fantastical comebacks do happen in sports. In fact, those moments are often the most famous and gratifying moments. Rather than try and predict

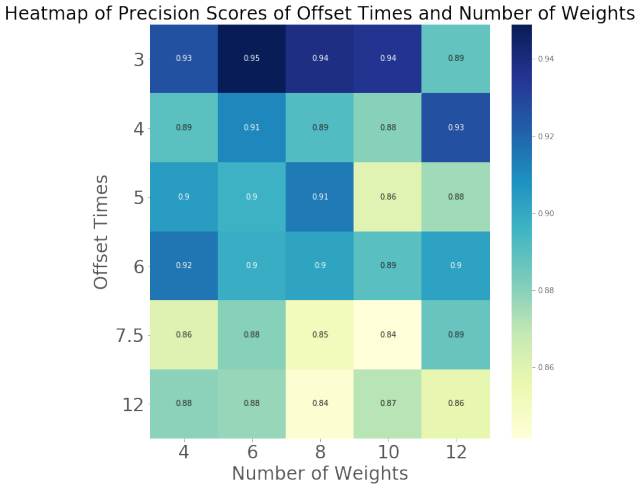


Figure 10: Game Prediction Precision Scores for Different Offset Times and Weight Numbers

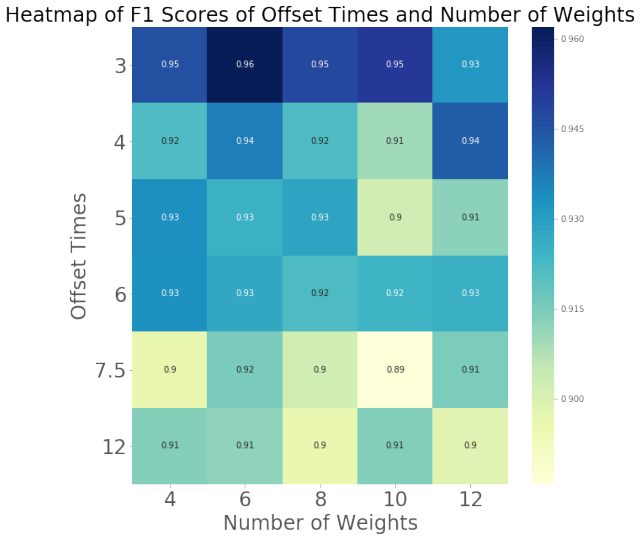


Figure 11: Game Prediction F1-Scores for Different Offset Times and Weight Numbers

every instance of every game, win modeling seeks to be an effective estimation method similar to how gradient descent can be used to find locally optimum solutions. Our probability calculations perform similarly to established models, and would most likely lead to the wrong conclusion if used on some of the edge cases such as Super Bowl 51. Instead, we use those estimated probabilities to test how early in a game we can predict the outcome accurately and how many samples of probabilities in the game we would need. This could be an interesting tool that could be referenced by the media and fans in conjunction with often cited win probabilities (i.e. teams with this probability of winning at this stage of the game win X% of the time). An interesting outcome of our results is that we can correctly classify almost 90% of games 12 minutes from the end.

This indicates that most games are decided soon after the fourth quarter begins, and maybe only 1 out of 10 games have exciting finishes. If all games had exciting endings, memorable games like Superbowl 51 would be the rule and not the exception.

4.4 Future Direction for Continued Research

If one wanted to contribute in this space there are several opportunities (but not limited too) that would advance either the algorithm or our understanding.

We sampled game data from 2009 until 2016 due to the availability of data. Football has been very data driven since the 1980's so more data exists than we have analyzed. Awareness of rule changes year to year would definitely need to be accounted for such as the addition of the two point conversion. One rule change that occurred in our dataset was the movement of kickoffs being downed in the end zone changing from the ball being spotted on the twenty yard line to being advanced to the twenty five yard line [2]. Using a stratified test train split would help condition on rule changes that may occur since part of your data will dwell in both the test and train datasets.

Perhaps the biggest deviation between our model and PFR's model is the final minutes of the game. As mentioned above, PFR uses a different model to examine end of half behavior. Future versions of our algorithm could definitely incorporate some temporal component where the model switches once a time threshold has been passed. This was outside the scope of our study but would certainly help us converge towards the PFR model if that was the goal.

Finally, probability modeling is humanity's best attempt at explaining the likelihood of an outcome occurring. We will never know the truth and this is our best effort to attempt to explain the behavior we are witnessing. The multitude of events ranging from presidential elections to championship sports games ending in a way that shocked the audience will continue on throughout human history. Perhaps the goal was never to actually predict more accurately, but merely to tell the narrative in such a way that the crowd treats the less likely outcome with a little more respect.

ACKNOWLEDGMENTS

The authors would like to thank Abhinav Maurya and Runshan Fu for providing insight and technical expertise as well as an abundance of patience.

REFERENCES

[1] ESPN Analytics. 2017. Charting the Patriots' incredible Super Bowl LI comeback. (2017). http://www.espn.com/.
[2] Judy Battista. 2016. New touchback rule: Another step toward eliminating kickoffs? (2016). http://www.nfl.com/news/story/0ap3000000648647/article/new-touchback-rule-another-step-toward-eliminating-kickoffs.
[3] Luke Benz. 2018. Improving College Basketball Win Probability Model. (2018). Yale Sports Analytics Group.
[4] Trey Causey. 2016. Building a Win Probability Model. (2016). http://thespread.us/building-a-win-probability-model-part-1.html.
[5] Maksim Horowitz. 2015. NFLscrapR. (2015).
[6] Josh Levin. 2017. We Are the 99 Percent. (2017).
[7] Dennis Lock and Dan Nettleton. 2014. Using random forests to estimate win probability before each play of an NFL game. (2014). http://homepage.divms.uiowa.edu/ dzimmer/sportsstatistics/nettletonandlock.pdf.
[8] Konstantinos Pelechrinis. 2017. iWinRNFL. (2017). https://arxiv.org/pdf/1704.00197.pdf.
[9] D.M.W POWERS. 2011. EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC. (2011). Journal of Machine Learning Technologies, School of Computer Science, Engineering and Mathematics, Flinders University.
[10] Pro-Football-Reference. 2017. The P-F-R Win Probability Model. (2017).
[11] Alan Ryder. 2004. Win Probabilities: a tour through win probability models for hockey. (2004). Hockey Analytics.
[12] Nate Silver. 2016. 2016 Election - FiveThirtyEight. (2016). fivethirtyeight.com/politics/elections/.
[13] American Statistician. 1991. On the Probability of Winning a Football Game. 3 (1991).
[14] StatsbyLopez. 2017. All Win Probability Models Are Wrong - Some Are Useful. (2017). statsbylopez.com/2017/03/08/all-win-probability-models-are-wrong-some-are-useful/.
[15] Sam Wang. 2016. Final Projections: Clinton 323 EV, 51 Democratic Senate Seats, GOP House. (2016). election.princeton.edu/2016/11/08/final-mode-projections-clinton-323-ev-51-di-senate-seats-gop-house/.