University of New Hampshire

University of New Hampshire Scholars' Repository

Honors Theses and Capstones | Student Scholarship

Spring 2019

# Predicting the Outcome of NFL Games Using Logistic Regression

Stephen Bouzianis
*University of New Hampshire, Durham*

Follow this and additional works at: https://scholars.unh.edu/honors

## Recommended Citation

Bouzianis, Stephen, "Predicting the Outcome of NFL Games Using Logistic Regression" (2019). *Honors Theses and Capstones*. 474.
https://scholars.unh.edu/honors/474

This Senior Honors Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Honors Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

Undergraduate Honors Thesis

# Predicting the Outcome of NFL Games Using Logistic Regression

by

Stephen Bouzianis

Pete T. Paul College of Business and Economics

University of New Hampshire

Advisor:

Dr. Ali Hojjat

Assistant Professor of Decision Sciences

May 2019

## Table of Contents

## 1. Introduction

The National Football League (NFL) is the highest grossing American professional sports league, generating over $13.5 billion revenue in 2017 (Statista, 2018). This multi-billion-dollar league garners an unrivaled interest from spectators around the United States, making it the most popular spectator sport in the country. Viewers around the country support their teams by purchasing game tickets, player merchandise, team gear, and by numerous other means. However, there is a new way in which the NFL stands to benefit from viewer consumption and action. The introduction of the legal sports gambling arena has provided the NFL and other US sports leagues with a future projected revenue stream that will boost their success emphatically. According to a 2018 Nielsen study, the NFL stands to gain an estimated $2.326 billion in annual revenue as a result of legal sports gambling (Garcia, 2018). This jump in revenue is due largely in part to the increase in consumption that forecasters are projecting as a result of people now having a larger rooting interest in games across the league. The additional interest in the NFL and other leagues will only increase the desire and need for those who can successfully predict the outcomes of their games.

Predicting the outcomes of NFL games is not a new activity, nor is it something that can be considered perfect science, but the importance and benefit cannot be understated. Engineering a prediction machine strong enough to reliably predict these outcomes would be of significance to consumers, NFL franchises, and the gambling industry alike. The unpredictability of the NFL has largely been attributed to the inability to measure or quantify the desire of the players on the field, or the number of intangibles that exist within each player, but that has not been a deterrent for those determined to understand which performance metrics and game attributes translate to victories.

The purpose of this research is to understand which statistics the best indicators of success are and use them to build a successful predictive model for determining the outcome of an NFL game. In particular, we explore the effectiveness and accuracy of Logistic Regression methodology in such prediction.

## 2. Literature Review

In this section we provide a brief review of the existing models in academic literature for predicting the outcome of sports games. We first focus on the models built using logistic regression, the technique we focus on in this paper, then we move on to other modeling techniques that are used to generate a similar prediction. The papers that use logistic regression particularly guide us in the set up and structure of our model, whereas those who explore other methods help us identify additional performance metrics that may be influential in our prediction and serve as benchmarks to compare the accuracy of our model.

### 2.1 Logistic Regression

Logistic regression is an extremely popular and quite successful technique for modeling the relationship between a number of independent variables and a dichotomous dependent variable (Kleinbaum, 2010, p.5). For problems in which one is trying to determine a probability of something happening, especially when there are only two possible outcomes (such as win or loss in a game, click or no click on an online ad, purchase or no purchase in response to an offered product, etc.), logistic regression is particularly appropriate and proven to be very powerful in various applications. The output from logistic regression is a value between 0 and 1. In many other types of models, such as linear regression, the output is not restricted to a value between 0 and 1, and so cannot be interpreted as a probability (Kleinbaum, 2010, p.6).

Willoughby (2002) used logistic regression to analyze and predict the outcomes of games in the Canadian Football League (CFL), a sport with extremely similar rules, statistics, and only a few differentiating factors from the NFL. The negligible differences between the sports are not significant enough to warrant ignoring the results and methods. The predictors chosen in this study were the head-to-head differences in a number of statistics. Rushing yardage differential, passing yardage differential, interception and fumble differential, and finally sack differential were the independent variables chosen. These indicators are meant to compare the two teams against one another and recognize who is superior in a specific category. Willoughby (2002) created three models, one for each of the three teams he wished to predict. The performance of each model was strong, but with some varying results. The three models returned 85.9%, 90.2%, and 78.8% correct prediction rates, with 90.2% being an exceptionally strong performance, while 78.8% being only relatively strong.

Shanahan (1984) performed a logistic regression model to predict the outcome of college basketball games at the University of Iowa. While basketball and football are two vastly different sports, the modeling techniques, the predictors used, and the results can all provide meaningful insight that help shape the direction of this paper. In her paper, many of the strongest predictors were those which translated to possession. Strong rebounding and strong turnover numbers were both highly influential, which is important to know when creating our model for NFL games. The accuracy of the models differed only slightly between the men's and women's game, with the success rates of 88% and 90%, respectively.

Kolbush and Sokol (2017) used logistic regression, together with a Markov chain model, to predict the outcome of the NCAA Football rankings. There are a number of major differences between the NFL and NCAA football games in terms of rules, scheduling, conferences, etc. but they are essentially the same game in terms of how they are played and the rules of the game. The general ideas and principles are the same with largely differing strategies, which allows for lessons to be taken from this paper. To account for the evolution of the game as well as the roster turnover, the authors decided to use 4 year rolling samples for their data. The Markov chain model was paired with logistic regression to create a ranking system, as opposed to a simple binary output. In our paper, we only seek a binary win/loss prediction. Kolbush and Sokol (2017) also examined whether or not point differentials should be used as one of the predictors. Point differentials can often be skewed one way or another; in some games a team will run up the score and inflate that point differential number, while sometimes the score does not indicate how close or lopsided a game could be. Using metrics outside of point differential would seem to create a story that is more accurate and reliable than relying on point differentials. Although the authors only generated a ranking of the teams and did not predict wins vs. losses, the success rate of their model was only near 61%.

## 2.2 Other Techniques

Aside from logistic regression, there are several other modeling techniques in sports prediction. Despite the differences in methods, there are many consistencies when considering the data preparation, feature selections, and other steps necessary to produce a successful model.

Warner (2010) used a Gaussian process predictive model in an attempt to outperform Vegas line makers, i.e., those who determine the betting lines in Las Vegas, when predicting the margin of victory in NFL games. This study examined features that are outside of the scope of what we normally associate with the outcomes of NFL games, such as weather condition differences between what a team typically plays in and the game they will play. Warner (2010) also performed feature selection to only include the most influential variables to predict success. Although the main goal of this study is to predict the margin of victory, he then proceeds to covert these margins to a win vs. loss prediction. The success rate of his model, excluding the point spread and margin of victory, is 64% percent.

Hamadani (2005) aimed to generate a better prediction than a human could, using machine learning. This paper also provided a comparison between different methodologies and modeling techniques such as logistic regression and Support Vector Machine (SVM). Surprisingly, the best performing model explored in this paper was one that employed logistic regression. The SVM results were similar but fell short of the accuracies generated using the regression analysis. Those accuracies were 65.83% for the 2005 season, 61.37% for the 2006 season and 67.08% for the 2007 season. Hamadani (2005) performed feature selection to generate a model that includes only the most influential variables. The overall success of the model was better than the compared accuracies of the 'expert' picks.

Silver (2014) runs a popular statistics website, FiveThirtyEight, which is well-known for its prediction machines and its relative accuracy for determining winners of all types. From presidential elections to sporting events, FiveThirtyEight has distinguished themselves as an industry leader in all things predictive in nature. The site uses a method that they developed that incorporates 'Elo', a rating system that they created from a variety of performance metrics to give a team a baseline rating, and then adjust based on their performance throughout the season. The 'average' team begins with an Elo rating of 1500 and most teams end the season with a rating between 1300 and 1700. These ratings are used as a baseline to predict the outcomes but are generated using a lot of the same ideas we explore in this paper.

Although logistic regression is an appropriate method for our problem, much of the existing work surrounding this topic has used other modeling methods. Neural networks and other types of models that fall under the machine learning umbrella have been used with strong success to predict both point spreads and outcomes of NFL games, but we consider those methodologies outside of the scope of this paper.

## 3. Research Objectives

The focus of this paper is to create an accurate predictive model that would predict the outcome of an NFL game with high accuracy. As an integral part of developing such a model, we also aimed to provide answers to the following questions:

- Which attributes/metrics/statistics are the most useful for predicting NFL outcomes?
- With what accuracy can we predict the winner of each game?

## 4. Data

The data used for this work is a comprehensive NFL dataset by Armchair Analysis, spanning from 2001 to 2018. This data is comprised of many individual tables that contain data about every single play, game, player and more for each of the 32 NFL teams. For each team and game, we utilized the following statistics:

- Home vs. Away (H)
- Points Scored (PTS)
- Rushing Yards (RY)
- Rushing Attempts (RA)
- Passing Yards (PY)
- Passing Attempts (PA)
- Pass Completions (PC)
- Sacks Against (SK)
- INT's for Defense (INTS)
- Fumbles Lost (FUM)
- Punts (PU)
- Punt Returns (PR)
- Punt Return Yards (PRY)

- Kick Returns (KR)
- Kick Return Yards (KRY)
- Penalty Yards (Against) (PEN)
- Time-of-Possession (TOP)
- Touchdowns (TD)
- Field Goals Made (FGM)
- Field Goals Attempted (FGAT)
- Drives in Red Zone (RZA)
- Red Zone Drive TD's (RZC)
- Starting Field Possession (SFPY)
- Net Punt Yardage (NPY)

The complete data dictionary to what is available at the time of this writing can be found online at: armchairanalysis.com/2019_AA_Table_Fields.pdf

## 5. Methodology

In this section we will elaborate on our logistic regression model, how we used Query Editor in Microsoft Excel along with R to drastically transform raw data into a workable state for regression analysis and discuss how we performed variable selection in conjunction with data partitioning and model validation to simplify our model and alleviate multicollinearity and overfit concerns.

### 5.1 Logistic Regression

Logistic regression is used widely to examine and describe the relationship between a binary response variable (e.g., 'success' or 'failure') and a set of predictor variables (G.M. Fitzmaurice, 2001). In our case, logistic regression is used to tell us whether or not a team would win (1) or lose (0). More specifically, the logistic regression model will return a probability between 0 and 1; this creates the need for a threshold of what will be considered a win or a loss. If the probability returned is greater than 0.5, then we would consider the prediction to be a win. The following mathematical equation represents the way in which the probability of a victory is determined:

$$\text{Prob}[Win] = \frac{1}{1 + e^{-U}}$$

where:

$$U = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots$$

and $x_i$ are the various independent variables (team performance metrics) chosen for the model.

The model coefficients, $\beta_i$, i.e., the weights by which the independent variables affect the outcome, are determined using Maximum Likelihood Estimation (MLE). For each variable (performance metric), the corresponding coefficient/weight can be interpret as follows: All else equal, each unit increase (decrease) in the variable $x_i$ will multiply (divide) the odds of winning, i.e., $\frac{\mathrm{Prob}[Win]}{\mathrm{Prob}[Loss]}$, by a factor of $e^{\beta_i}$. Therefore, a positive (negative) coefficient, indicates a positive (negative) relationship between that variable (performance metric) and the odds of winning. For further explanation on logistic regression, see Fitzmaurice (2001).

There are several ways we could use logistic regression: building a separate model for each pair of teams; a separate model for each team (against all others), or one model that explains all teams. We chose to use a separate model for each of the 32 NFL franchises because the performance metrics that best explain one team's success may not be optimal for another team. Therefore, a single model may not be able to accurately predict the outcome of games for all teams. On the other hand, fitting a separate model for each pair of teams would not be practical since a particular pair does not play against each other too often for us to have enough data points to fit a regression model properly. Creating a separate model for each of the teams strikes a balance between tailoring the models to unique attributes of each team, while having enough data to support the fitting and testing of each model. This approach is also consistent with the literature, for instance, the work of Willoughby (2002).

## 5.2 Data Preparation

Before the regression could be performed there were a number of data preparation measures taken in order to merge, clean, preprocess, and transform the raw data into the necessary structure to carry out the analysis.

The relevant data fields (i.e., team performance metrics) that we were interested in using as our independent variables and the results of the games (which would be the dependent variables in our model), were spread across various tables. Using Query Editor in

Microsoft Excel, we were able to gather and merge all relevant data fields in to one table. This table showed, for every single game from 2001 to 2018, the season/year and game ID, home and away team IDs, a binary win/loss column (i.e., our dependent variable) indicating whether the home team won the game, and finally all of the performance metrics that we mentioned before in our Data section as realized during that game. This data was then loaded in R Studio for the remainder of transformations.

There were three teams that required extra consideration due to either changing locations, or the expansion of the NFL. The Houston Texans were an expansion team prior to the 2002 season, thus not having a full season of data that the other teams did. For both the St. Louis Rams and the San Diego Chargers, who both moved to Los Angeles before the 2016 and 2017 seasons, respectively, we considered the teams to be the same before and after the move. To account for the difference in names, we assigned the team name of SDLA for the Chargers, and STLA for the Rams. We did not omit those teams from consideration or take any other steps to manipulate their data specifically and simply assumed they were the same team.

The classification model should not use the performance result of a game after it has happened to predict the outcome of that same game. The input variables to the model on any particular game should only be based on the performance of the two teams as observed up to and prior to that game. Therefore, the merged data is still not ready for fitting the model. In order to correct this, we wrote an R script to calculate 5-game moving averages (MA) and all-time cumulative averages (CA) of each metric up to (and excluding) each game. The MA and CA of each statistic became the initial candidates for independent variables. The very first 4 games played by each team in the history of our data were eliminated from the analysis due to not having a 5-game MA metric yet.

Our processed data table had the following structure: [Season, Year, Game ID, Home team ID, Visiting team ID, Home team win status (binary), Home team MA & CA statistics, Visitor team MA & CA statistics].

## 5.3 Feature Generation

Feature generation is a common step in predictive modeling in which non-linear transformations of variables are created to form new additional independent variables. This would allow the model to take advantage of possible nonlinear relationships between an input and the output which can often drastically improve the prediction accuracy. Simple linear transformations would only create an undesired multicollinearity in the model and would therefore be futile. It is hard to guess what specific transformations and on which variables would help increase the prediction accuracy, therefore, we choose to create as many transformations as we can think of and apply them to all variables, and let the variable section step, explained in the following section, to then trim the model down to only those variables that are significant.

In addition to the original moving and cumulative averages of teams' performance metrics, we wrote an R script to apply the following non-linear transformations to those MA and CA values and used them as additional independent variables: square, square root, logarithm, pairwise ratios and products. These transformations were not possible on every performance metric, e.g., some leaded to frequent division by zeros, and such cases were not generated in this process. Having these additional variables gave us a total of 505 independent variables to consider for our model.

### 5.4 Feature Selection

Feature selection is often used in analytics as an easy and effective way of excluding variables that do not contribute, in a statistically significant manner, to the predicted outcome; thereby retaining only the most important metrics in the model. There are three main advantages in performing feature selection: i) having a smaller/simpler model to work with which would require less data to collect as input, ii) alleviating multicollinearity issues among independent variable, and iii) alleviating overfit issues, which could easily happen following the feature generation step. We will elaborate more on overfit issues in the next section.

There are a number of standard methodologies for performing variable selection which we explored to determine which would be appropriate for our case. The best subsets method tests every possible subset of variables returns the best model at any given number of predictors. Backwards elimination begins with all variables included in the model and iteratively eliminates the least impactful variable, one at a time, until the model is reduced to a desired number of predictors, or all the remaining variables are statistically significant. Similarly, forward selection starts with only the intercept and iteratively adds the most impactful variable, one at a time, until there is no significant benefit from continuing. Lastly, sequential replacement replaces once variable at a time to check if any other possible variable would enhance the performance of that model.

The Best Subsets method is optimal, but it is also impractical for models that consider more than 20 variables as there would be a prohibitive number of variable subsets to fit and test. When comparing forward and backwards, we found that the final reduced models found through backwards elimination had a higher accuracy in our application. We then performed backward elimination down to 80 and then 20 variables and found that having about 20 variables in the model is the sweet spot for simplicity of the model, high accuracy, and having resolved most overfit concerns. These steps were looped in our R Script to produce models of both sizes for each of the 32 NFL franchises, with predictors that perform best for that team, independent of the others.

## 5.5 Data Partitioning and Model Validation

The primary goal of any predictive model is to produce an accurate prediction on new data which the model has not seen before. Consequently, only looking at the accuracy of the model within the same dataset that was used to fit the model is not sufficient for determining how well the model performs. Instances in which a model performs well on the dataset that was used to fit the model but does not perform well on new data is referred to as overfit. Excessive feature generation and including too many variables in a model can lead to significant overfit issues.

A common way to validate the model and ensure that there is not too much overfit is to partition the data into a training set and a testing set. The training data is used to fit the model, i.e., estimate the regression coefficients, while the test data is used to confirm that the model produces consistent accuracy rates when faced with new data. It is expected and normal that a model would perform better on the training data, since the model coefficients are particularly optimized to achieve that, however, we do not want the performance on test data to be drastically worse. When a model has overfit issues, variable selection is commonly used to reduce the number of features considered for the model and correct the overfitting.

We used the 2001-2016 data for training our models and held the 2017 and 2018 data for validation. This partitioning left us with an average of 300 games per team for training (i.e., fitting) each model and about 33 games for validation. This threshold was set in R studio and this process was made as an integral part of our R script loop for creating and testing the models for each of the 32 teams.

# 6. Results

In this section we will discuss the results of our models, with a section showing the various prediction accuracies, and a section highlighting the most commonly used predictors across the models.

## 6.1 Prediction Accuracy

The following table shows the prediction accuracies obtained for each team using a large model with 80 variables, and then a small model with only 20 variables, each measured on both training (2001-2016) and validation (2017-2018) data. These results are sorted in descending order by the accuracy of our test data from our small model. For a complete list of all coefficient information, please see the Appendix.

The results, for the most part, align with the consistency of the teams since the beginning of our dataset. Teams who were either consistently good, like the New England Patriots, or teams that were consistently bad, like the Cleveland Browns, had the best accuracies across the models. Similarly, teams that have been inconsistent from year to year, like the Carolina Panthers and the New Orleans Saints, had lower accuracy totals relative to the others.

| | Large Model Accuracy (80 Vars) | | Small Model Accuracy (20 Vars) | |
|---|---|---|---|---|
| Team | Train | Test | Train | Test |
| New England Patriots (NE) | 0.900 | 0.500 | 0.823 | 0.763 |
| Cleveland Browns (CLE) | 0.944 | 0.781 | 0.787 | 0.719 |
| New York Jets (NYJ) | 0.864 | 0.688 | 0.731 | 0.688 |
| Kansas City Chiefs (KC) | 0.847 | 0.457 | 0.726 | 0.657 |
| Dallas Cowboys (DAL) | 0.887 | 0.471 | 0.715 | 0.647 |
| Chicago Bears (CHI) | 0.915 | 0.697 | 0.727 | 0.636 |
| San Franciso 49ers (SF) | 0.928 | 0.313 | 0.733 | 0.625 |
| Oakland Raiders (OAK) | 0.916 | 0.563 | 0.756 | 0.625 |
| Indianapolis Colts (IND) | 0.948 | 0.485 | 0.769 | 0.606 |
| Pittsburgh Steelers (PIT) | 0.848 | 0.545 | 0.759 | 0.606 |
| Green Bay Packers (GB) | 0.900 | 0.500 | 0.748 | 0.594 |
| Minnesota Vikings (MIN) | 0.883 | 0.471 | 0.734 | 0.588 |
| Detroit Lions (DET) | 1.000 | 0.531 | 0.772 | 0.563 |
| Houston Texans (HOU) | 1.000 | 0.500 | 0.773 | 0.563 |
| Miami Dolphins (MIA) | 0.860 | 0.516 | 0.708 | 0.548 |
| Seattle Seahawks (SEA) | 0.907 | 0.485 | 0.740 | 0.545 |
| Baltimore Ravens (BAL) | 0.907 | 0.515 | 0.730 | 0.545 |
| Jacksonville Jaguars (JAC) | 0.926 | 0.657 | 0.733 | 0.543 |
| Atlanta Falcons (ATL) | 0.910 | 0.529 | 0.752 | 0.529 |
| Philadelphia Eagles (PHI) | 0.846 | 0.595 | 0.723 | 0.486 |
| Tennessee Titans (TEN) | 0.908 | 0.529 | 0.744 | 0.471 |
| Cincinati Bengals (CIN) | 0.938 | 0.406 | 0.710 | 0.469 |
| Buffalo Bills (BUF) | 1.000 | 0.273 | 0.727 | 0.455 |
| Washington Redskins (WAS) | 0.941 | 0.452 | 0.717 | 0.452 |
| Arizona Cardinals (ARI) | 0.865 | 0.313 | 0.684 | 0.438 |
| Denver Broncos (DEN) | 0.849 | 0.563 | 0.727 | 0.406 |
| Carolina Panthers (CAR) | 0.904 | 0.576 | 0.762 | 0.394 |
| San Diego/Los Angeles Chargers (SDLA) | 0.887 | 0.382 | 0.764 | 0.353 |
| St. Louis/Los Angeles Rams (STLA) | 0.930 | 0.667 | 0.733 | 0.333 |
| Tampa Bay Buccaneers (TB) | 0.813 | 0.406 | 0.713 | 0.313 |
| New Orleans Saints (NO) | 0.905 | 0.278 | 0.669 | 0.278 |
| New York Giants (NYG) | 0.901 | 0.406 | 0.691 | 0.250 |

Examining the differences between the "Large Model" and "Small Model" accuracies, it is apparent that there was significant overfitting for the larger model. For example, when we look at the outputs for San Francisco (SF), the training accuracy for the "Full Model" is 92.8% and the testing accuracy is 31.3%. This clearly shows that the model was overfit to the training data and did not perform nearly as well on the test data. After feature selection was used to trim the number of variables further down to 20, those same accuracy outputs are much closer to one another, at 73.3% and 62.5%, respectively. This means we can now be confident that the smaller model delivers a similar level of accuracy when applied to new data, whereas the large model with 80 variables cannot be trusted on new data. For some of the teams, particularly in the bottom half of the table, there is still a large gap between the train and test accuracies, which suggests a persistent overfit which could demand further variable elimination, maybe down to 10 variables. But in our tests, further variable elimination did not help with correcting the overfit in a noticeable way; It appears as if the recent performance of those teams during 2017-2018 simply could not be accurately explained based on their historic performance prior to 2017.

## 6.2 Statistically Significant Performance Metrics

Because we created a separate model for each team and performed feature selection independently for each of the 32 NFL franchises, there were different variables that were deemed statistically significant in predicting the probability of a win for each specific team. We combined the list of variables across all 32 (small) models, and using a pivot table in Excel, we generated a list of those most commonly found variables:

- Cumulative average of SFPY
- Cumulative average of PTS
- Squared cumulative average of SFPY
- Cumulative average of TOP
- Home or Away status (binary)
- Cumulative average of FGAT
- Cumulative average of PA
- Cumulative average of RY
- Squared cumulative average of PR
- Cumulative average of PR
- Square root of the cumulative average of PTS
- Squared cumulative average of TDs
- Squared cumulative average of PTS
- Square root of the cumulative average of PU
- Square root of the cumulative average of SFPY
- Cumulative average of KRY
- Cumulative average of TDs

The number of times that each of these predictors shows up among the 32 models is not drastically different from others, so we listed all variables that were considered a significant predictor for at least 7 of the 32 teams.

It is worth noting that none of the metrics listed above relate to the opposing team, nor are there any moving average metrics that made the list. They did appear in some models but were not consistently included. See the Appendix for a complete list of the variables used in each individual model.

Finally, we should point out that many of these variables are the result of our feature generation step, particularly through squaring or taking the square root of an original variable. It would have been hard to "guess" that such metrics would be significant predictors of a game outcome by intuition.

## 7. Conclusions and Future Work

We explored the effectiveness and accuracy of Logistic Regression methodology in predicting the outcome of NFL games. Our process began with transforming large amount of raw data, obtained from Armchair Analysis, into a format that was suitable for logistic regression. We wrote an R Script that applied moving averages and cumulative averages to each team performance metric we considered as independent variable for both the home and opposing teams, and then applied a number of non-linear transformations, which left us with a total of 505 independent variables to start the regression analysis. We then performed backward feature selection to trim those variables down to 80 and 20 for our two different sized models, while partitioning out data into training and validation sets to check for model overfit. After the previous steps were taken, various accuracy statistics were generated to show how well our models performed. Teams that were more consistent, either good or bad, like New England and Cleveland were predicted with the highest accuracies, above 70% on both training and test data. Teams that were more inconsistent over the same timeframe, like New Orleans and the New York Giants, showed near 70% accuracy on training data, but had the low accuracies near 30% on test data, which could be improved further through variable selection.

As mentioned in the introduction, one key limitation of any sports prediction is that it is impossible to measure the amount of desire a given player or team has. We try to assign value to all other measurables and metrics on the field, but often times the intangibles within players is the difference between a win and a loss. Furthermore, we left out the environmental aspects of the games (such as stadium temperature and humidity) from our predictors, because our dataset contained too many missing values in those respects. Additionally, we did not explore an exhaustive list of non-linear transformations to generate our additional features. Exploring other transformations may help improve the accuracy further.

After comparing our results with those from previous works, it appears that more successful models can be generated using other modeling techniques. In this project, we did not explore other classification techniques such as neural networks, k-nearest neighbors or SVM. The viability of different classification methods is highly problem-specific and dependent on the data; therefore, any of those alternative methods may provide a stronger result. Given more time and a stronger understanding of additional methods, it would be valuable to test and see how those alternative techniques would compare to our models.

As the gambling industry grows and becomes more widely legalized, we expect that there would be a strong demand for predicting whether or not a team will cover a certain point spread against the opposing team. Being able to predict the point spread, instead of a binary win/loss, would enhance the value of our models to the gambling community, but in that case, we would need to use a method that produces a continuous output, like linear regression. In fact, all the data preparation and transformations that we performed in this study, as well as the data partitioning and model validation steps, would directly apply to the case of linear regression. The only difference would be that the point spread observed in each game would become the dependent variable, the model coefficients would be estimated using the Least Squares method, and the accuracy of the models would be measured using average distance metrics such as Mean Absolute Deviation (MAD), Mean Absolute Percent Error (MAPE), or Root Mean Square Error (RMSE).

While we certainly cannot quantify the heart of an athlete or foresee unpredictable events such as injuries, that did not deter us from attempting to search for and test available performance metrics which may be crucial indicators of a game's outcome. Our final model and findings provide the groundwork and template for those attempting to use logistic regression to predict outcomes of other sporting events. In fact, some of the performance metrics we discover to be the most indicative of winning football had not been tested before in prior academic literature.

Moving forward, it would be interesting to apply the same techniques used in this paper to other sports. Much of the data preparation and modeling we performed could be applied in a similar fashion to NBA, NHL, or MLB games. It is possible that logistic regression is more suitable for a different sport than football, where there are more games played in a season and thus higher volume of data available to train and test the models. Having only 16 games in the regular season and a maximum of 4 additional playoff games, there are not nearly as many records in NFL as there would be in a baseball season of 162 games.

## References


Armchair Analysis Comprehensive NFL Data, Retrieved February 2019 from: www.armchairanalysis.com/data.php.

Fitzmaurice, G. M., and N. M. Laird. (2001) *Multivariate Analysis: Discrete Variables (Logistic Regression*). International Encyclopedia of the Social & Behavioral Sciences, pp. 10221-10228. Retrieved from https://doi.org/10.1016/B0-08-043076-7/00476-9

Garcia, Nicholaus. (2018) *Breaking Down The NFL's Billion Dollar Benefits From Legalized Sports Betting*. Retrieved from https://www.playusa.com/nfl-legal-sports-betting-revenue

Hamadani, Babak. (2005) *Predicting the outcome of NFL games using machine learning*. Retrieved from http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf.

Willoughby, Keith A. (2002) *Winning Games in Canadian Football: A Logistic Regression Analysis*, The College Mathematics Journal, 33:3, 215-220

Kleinbaum D.G., Klein M. (2010) *Introduction to Logistic Regression*. Logistic Regression. Statistics for Biology and Health. Springer, New York, NY

Kolbush, J., and J. Sokol. (2017) *A Logistic Regression/Markov Chain Model for American College Football*. International Journal of Computer Science in Sport 16, no. 3: 185-196.

Shanahan, Kathleen J. (1984) *A model for predicting the probability of a win in basketball.* M.A. Thesis. University of Iowa. Retrieved from https://ir.uiowa.edu/etd/5082

Silver, Nate. (2014) *Introducing NFL Elo Ratings.* FiveThirtyEight. https://fivethirtyeight.com/features/introducing-nfl-elo-ratings/

Statista. (2018). National Football League (NFL) Statistics & Facts. Retrieved from https://www.statista.com/topics/963/national-football-league

Warner, Jim. (2010) *Predicting margin of victory in NFL games: Machine learning vs. the Las Vegas line*. Technical Report.

# Appendix: Coefficient Information and Confusion Matrices for Individual NFL Teams

In this section, we will show the individual outputs for each model we created, as well as the confusion matrices associated with each. The variable names below are all comprised of performance acronyms defined in the 'Data' section 4, followed by either MA meaning the 5-game moving average, or CA meaning the cumulative average. A number "2" in the variable name indicates that this was a performance metric from the opposing team. Any prefix denotes the nonlinear transformation performed on the variable before its inclusion in the model, as introduced in the Feature Generation section 5.3: "Sq" means the squared term, "Sqrt" means the square root of the term, "log" means the logarithm of the term, "mult" means the pairwise product of the term, with "rat" meaning pairwise ratio of the term.

**San Francisco 49ers**

```
Coefficients:
                   Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   -6.814123e+03 3.320187e+03 -2.052331 4.013748e-02
ra_MA         -1.790999e-01 6.108105e-02 -2.932168 3.366042e-03
top_MA         3.023449e-01 9.848084e-02  3.070088 2.139954e-03
sfpy_MA        1.998080e+00 1.079014e+00  1.851765 6.405952e-02
pts_CA        -1.875725e+00 6.129979e-01 -3.059921 2.213951e-03
py_CA          1.143662e+00 4.517209e-01  2.531790 1.134819e-02
pry_CA        -8.788293e+01 2.300658e+01 -3.819905 1.335032e-04
fgat_CA       -7.841214e+00 3.420111e+00 -2.292678 2.186653e-02
npy_CA        -8.448213e-01 3.418267e-01 -2.471490 1.345512e-02
ry_MA2        -1.423323e-02 5.243727e-03 -2.714335 6.640897e-03
sfpy_MA2       7.613202e-03 3.287280e-03  2.315958 2.056057e-02
sfpy_CA2      -3.591757e+01 1.794156e+01 -2.001920 4.529335e-02
Sq_sfpy_MA    -9.754495e-04 5.227317e-04 -1.866061 6.203277e-02
Sq_py_CA      -2.504012e-03 9.354697e-04 -2.676743 7.434166e-03
Sq_pr_CA      -1.976662e+00 1.065976e+00 -1.854321 6.369322e-02
Sq_pry_CA      1.099456e+00 2.739586e-01  4.013221 5.989571e-05
Sq_npy_CA      2.384951e-03 9.827355e-04  2.426849 1.523058e-02
Sq_sfpy_CA2    1.627145e-02 8.096936e-03  2.009582 4.447550e-02
Sqrt_sfpy_MA  -4.906168e+01 2.653407e+01 -1.849007 6.445679e-02
Sqrt_pry_CA    4.250272e+02 1.138533e+02  3.733112 1.891286e-04
Sqrt_sfpy_CA2  9.169754e+02 4.594690e+02  1.995729 4.596347e-02

Confusion Matrices:

Train:

p_data   0   1
     0 115  41
     1  33  88

Test:

p_data  0  1
     0 15  5
     1  7  5
```

**Atlanta Falcons**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   1.993485e+05 6.093261e+04  3.271623 1.069321e-03
pr_MA        -8.051473e-01 2.000572e-01 -4.024586 5.707563e-05
rza_MA       -8.477081e+00 2.784479e+00 -3.044405 2.331412e-03
ints_CA      -2.596231e+02 1.230293e+02 -2.110254 3.483650e-02
pr_CA         2.324152e+02 1.273056e+02  1.825649 6.790319e-02
kr_CA        -2.476361e+02 6.640297e+01 -3.729293 1.920174e-04
td_CA        -4.341525e+02 1.010956e+02 -4.294476 1.751066e-05
rzc_CA        6.466145e+01 1.786002e+01  3.620458 2.940823e-04
sfpy_CA       1.112997e+03 3.426770e+02  3.247947 1.162408e-03
Sq_ints_CA    1.089769e+02 5.818795e+01  1.872843 6.109006e-02
Sq_pr_CA     -5.161585e+01 2.906038e+01 -1.776159 7.570673e-02
Sq_sfpy_CA   -5.132547e-01 1.602018e-01 -3.203801 1.356262e-03
Sqrt_rza_MA   3.062093e+01 9.852463e+00  3.107947 1.883921e-03
Sqrt_ra_CA   -4.622948e+01 1.010425e+01 -4.575252 4.756476e-06
Sqrt_pa_CA    6.959849e+01 2.128880e+01  3.269254 1.078313e-03
Sqrt_pc_CA   -6.794042e+01 1.935634e+01 -3.509983 4.481347e-04
Sqrt_kr_CA    1.015286e+03 2.707522e+02  3.749873 1.769239e-04
Sqrt_td_CA    1.203184e+03 2.811870e+02  4.278944 1.877821e-05
Sqrt_rzc_CA  -7.876735e+01 2.479704e+01 -3.176482 1.490732e-03
Sqrt_sfpy_CA -2.821570e+04 8.624997e+03 -3.271387 1.070214e-03
Sqrt_pts_MA2 -1.185779e+00 2.784878e-01 -4.257923 2.063350e-05

Confusion Matrices:

Train:

p_data   0   1
     0  96  31
     1  38 113

Test:

p_data  0  1
     0  0  0
     1 16 18
```

**Jacksonville Jaguars**

```
Coefficients:
                 Estimate   Std. Error   z value     Pr(>|z|)
(Intercept) -4.118014e+03 1.741090e+03 -2.365193 0.0180207009
ry_MA        1.647929e+00 5.073102e-01  3.248365 0.0011607018
kry_MA       3.712587e-02 1.057365e-02  3.511170 0.0004461392
top_MA      -3.469286e+02 1.197358e+02 -2.897452 0.0037620707
pts_CA      -4.294714e+01 1.082238e+01 -3.968364 0.0000723677
sk_CA       -1.463939e+01 9.952978e+00 -1.470855 0.1413302140
fum_CA       1.682451e+02 5.457675e+01  3.082724 0.0020511511
kr_CA       -3.232929e+02 9.222349e+01 -3.505538 0.0004556857
fgat_CA     -1.721349e+01 5.896504e+00 -2.919270 0.0035085226
kr_MA2       7.234229e-01 1.949601e-01  3.710620 0.0002067526
fgm_MA2     -4.578858e+00 1.470074e+00 -3.114712 0.0018412475
Sq_ry_MA    -2.325502e-03 7.152547e-04 -3.251292 0.0011488181
Sq_top_MA    1.903685e+00 6.642660e-01  2.865848 0.0041589393
Sq_pts_CA    1.005112e+00 2.598174e-01  3.868531 0.0001094929
Sq_sk_CA     1.736280e+00 1.579601e+00  1.099189 0.2716855774
Sq_fum_CA   -9.962653e+01 3.394486e+01 -2.934952 0.0033359935
```

```
Sq_kr_CA     4.305161e+01 1.239920e+01  3.472128 0.0005163495
Sq_sfpy_CA   1.437459e-04 4.496321e-05  3.196967 0.0013888071
Sq_fgm_MA2   1.405995e+00 4.621360e-01  3.042383 0.0023471294
Sqrt_ry_MA  -2.317750e+01 7.201775e+00 -3.218304 0.0012895123
Sqrt_top_MA  2.543740e+03 8.737363e+02  2.911336 0.0035988674
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 135  42
     1  30  63
```

Test:

```
p_data  0  1
     0 13 11
     1  5  6
```

**Cleveland Browns**

Coefficients:

```
                 Estimate   Std. Error    z value     Pr(>|z|)
(Intercept) -2.140464e+02 2.152620e+02 -0.9943529 0.3200511011
pts_CA      -2.293736e+01 2.909610e+01 -0.7883309 0.4305031795
kry_CA       1.559315e-01 6.118222e-02  2.5486411 0.0108143535
top_CA      -4.590509e+00 1.626202e+00 -2.8228400 0.0047600327
td_CA        2.131094e+02 1.861489e+02  1.1448328 0.2522784177
sk_MA2      -3.755335e+00 1.403706e+00 -2.6753001 0.0074662393
ra_CA2      -1.494119e+01 5.067453e+00 -2.9484614 0.0031935998
pr_CA2       1.389986e+01 1.843941e+01  0.7538127 0.4509617099
fgat_CA2    -3.501817e+00 1.625189e+00 -2.1547142 0.0311842056
Sq_rzc_MA2  -3.234472e-01 1.086786e-01 -2.9761806 0.0029186291
Sq_top_CA2   1.227117e-02 4.533972e-03  2.7064950 0.0067997611
Sq_sfpy_CA2  5.024277e-05 2.050467e-05  2.4503089 0.0142733708
Sqrt_pc_MA   5.555754e-01 4.396812e-01  1.2635868 0.2063783831
Sqrt_pts_CA  1.193874e+02 2.262902e+02  0.5275854 0.5977871614
Sqrt_pu_CA  -5.402949e+01 2.618347e+01 -2.0634967 0.0390654615
Sqrt_td_CA  -4.102422e+02 4.713867e+02 -0.8702879 0.3841430931
Sqrt_rza_CA  1.991409e+01 6.012423e+00  3.3121566 0.0009257971
Sqrt_npy_CA  7.042812e+00 3.403981e+00  2.0689929 0.0385467531
Sqrt_sk_MA2  1.248547e+01 4.337464e+00  2.8785188 0.0039954748
Sqrt_ra_CA2  1.572867e+02 5.369977e+01  2.9290006 0.0034005373
Sqrt_pr_CA2 -5.132402e+01 5.593060e+01 -0.9176376 0.3588086349
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 167  41
     1  16  43
```

Test:

```
p_data  0  1
     0 23  7
     1  2  0
```

**Philadelphia Eagles**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   4.446042e+00 9.723650e+01  0.045724 9.635302e-01
pu_MA         7.285198e+01 3.722323e+01  1.957164 5.032818e-02
npy_MA       -2.045701e+00 8.641274e-01 -2.367361 1.791545e-02
pa_CA         1.650732e+00 5.955338e-01  2.771854 5.573812e-03
ints_MA2      1.936807e+00 1.117671e+00  1.732896 8.311419e-02
ra_CA2       -4.052167e-01 1.706710e-01 -2.374256 1.758436e-02
top_CA2       1.187766e+01 5.607425e+00  2.118202 3.415798e-02
Sq_py_MA      2.294813e-05 1.050957e-05  2.183545 2.899570e-02
Sq_pu_MA     -2.488601e+00 1.268410e+00 -1.961984 4.976438e-02
Sq_npy_MA     1.941140e-03 7.947535e-04  2.442443 1.458825e-02
Sq_pts_MA2   -1.309755e-03 6.000363e-04 -2.182793 2.905102e-02
Sq_ints_MA2  -7.198330e-01 4.874184e-01 -1.476828 1.397219e-01
Sq_top_CA2   -1.919164e-01 9.238919e-02 -2.077260 3.777754e-02
Sq_rzc_CA2    1.008923e+00 3.735308e-01  2.701044 6.912218e-03
Sqrt_pu_MA   -2.120414e+02 1.086578e+02 -1.951460 5.100231e-02
Sqrt_npy_MA   3.579390e+01 1.536035e+01  2.330279 1.979140e-02
Sqrt_py_CA   -8.661804e+00 2.082220e+00 -4.159889 3.184024e-05
Sqrt_kr_CA    1.064122e+01 4.697897e+00  2.265103 2.350635e-02
Sqrt_rzc_CA   9.459861e+00 2.991465e+00  3.162283 1.565373e-03
Sqrt_sfpy_CA -6.205313e+00 1.824466e+00 -3.401166 6.709911e-04
Sqrt_pc_CA2  -4.836417e+00 1.524532e+00 -3.172395 1.511875e-03

Confusion Matrices:

Train:

p_data   0   1
     0  74  33
     1  46 132

Test:

p_data  0  1
     0  4 12
     1  7 14
```

**Dallas Cowboys**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -2.515211e+02 1.182695e+02 -2.126678 3.344688e-02
pry_MA       -1.563715e+00 6.899289e-01 -2.266487 2.342161e-02
ra_CA         3.997774e+00 1.261799e+00  3.168314 1.533259e-03
top_CA       -9.942917e+00 2.747515e+00 -3.618877 2.958842e-04
fgat_CA       4.468933e+00 2.845115e+00  1.570739 1.162433e-01
rza_CA        1.354195e+01 4.977818e+00  2.720458 6.519154e-03
fum_MA2       1.353033e+00 4.323827e-01  3.129248 1.752545e-03
kr_MA2       -4.705128e-01 1.880344e-01 -2.502270 1.233999e-02
ra_CA2       -1.601564e+01 4.815720e+00 -3.325701 8.819648e-04
pa_CA2       -5.150535e-01 1.747318e-01 -2.947680 3.201678e-03
fum_CA2      -4.181784e+00 1.835873e+00 -2.277818 2.273740e-02
Sq_pry_MA     1.155055e-02 5.387551e-03  2.143932 3.203830e-02
Sq_pa_CA      4.215639e-02 1.625601e-02  2.593280 9.506537e-03
Sq_td_CA     -1.250630e+00 5.379224e-01 -2.324927 2.007585e-02
Sq_py_CA2     1.534709e-04 4.127823e-05  3.717962 2.008365e-04
Sqrt_pry_MA   9.421229e+00 4.062424e+00  2.319115 2.038879e-02
Sqrt_rza_CA  -1.331608e+01 6.825976e+00 -1.950795 5.108143e-02
```

```
Sqrt_ra_MA2  -1.710954e+00 4.752524e-01 -3.600095 3.181004e-04
Sqrt_py_MA2  -3.810650e-01 1.138816e-01 -3.346152 8.194137e-04
Sqrt_pts_CA2 -4.582284e+00 1.083702e+00 -4.228359 2.354015e-05
Sqrt_ra_CA2   1.643931e+02 4.995098e+01  3.291089 9.980021e-04
```

Confusion Matrices:

Train:

```
p_data   0   1
     0  90  35
     1  43 106
```

Test:

```
p_data  0  1
     0  2  0
     1 12 20
```

**New York Jets**

Coefficients:

```
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -4.884714e+02 2.266496e+02 -2.155183 3.114748e-02
rza_MA        6.161815e+01 1.967014e+01  3.132572 1.732818e-03
pts_CA        2.292248e+01 1.981173e+01  1.157015 2.472661e-01
pa_CA         1.476794e+01 6.853572e+00  2.154780 3.117903e-02
pry_CA        7.845661e-01 2.833968e-01  2.768437 5.632592e-03
rza_CA       -1.859419e+02 4.913983e+01 -3.783934 1.543686e-04
sfpy_CA      -1.582974e-01 5.297095e-02 -2.988381 2.804594e-03
py_CA2       -3.639402e-01 1.738235e-01 -2.093734 3.628365e-02
ints_CA2      2.578648e+00 1.077276e+00  2.393673 1.668061e-02
Sq_pr_MA      1.949860e-01 4.730901e-02  4.121541 3.763468e-05
Sq_rza_MA    -2.873572e+00 1.007051e+00 -2.853452 4.324705e-03
Sq_pts_CA    -5.800449e-01 4.929710e-01 -1.176631 2.393429e-01
Sq_pa_CA     -2.366490e-01 1.052506e-01 -2.248433 2.454858e-02
Sq_kr_CA      1.132325e+00 2.834764e-01  3.994425 6.485131e-05
Sq_rza_CA     2.064722e+01 5.184052e+00  3.982834 6.809832e-05
Sq_pa_MA2     1.045046e-03 4.983564e-04  2.096986 3.599485e-02
Sqrt_rza_MA  -1.543278e+02 4.718014e+01 -3.271033 1.071553e-03
Sqrt_rza_CA   2.680241e+02 7.522158e+01  3.563128 3.664612e-04
Sqrt_td_MA2  -2.271113e+00 6.456549e-01 -3.517535 4.355755e-04
Sqrt_py_CA2   1.089804e+01 5.127081e+00  2.125583 3.353797e-02
Sqrt_fgm_CA2 -6.955719e+00 3.002349e+00 -2.316759 2.051688e-02
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 112  42
     1  33  92
```

Test:

```
p_data  0  1
     0 17  4
     1  6  5
```

**Green Bay Packers**

```
Coefficients:
                 Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)   2.287190e+03 7.627453e+02  2.9986289 2.711974e-03
Home.1        1.125077e+00 2.982138e-01  3.7727199 1.614775e-04
pc_MA         2.365230e-01 7.716012e-02  3.0653525 2.174136e-03
ra_CA        -3.709930e+00 9.101824e-01 -4.0760286 4.581139e-05
pa_CA        -1.499144e+02 4.302938e+01 -3.4840012 4.939772e-04
td_CA        -2.037046e+02 7.961594e+01 -2.5585903 1.050975e-02
fgat_CA      -6.172922e+01 6.032357e+01 -1.0233019 3.061651e-01
rza_CA       -2.969577e+01 9.638951e+00 -3.0808091 2.064389e-03
rza_MA2       2.611292e+00 1.271882e+00  2.0530925 4.006361e-02
Sq_rza_MA    -8.685929e-02 3.972327e-02 -2.1866098 2.877102e-02
Sq_sfpy_MA    1.377691e-05 6.106055e-06  2.2562701 2.405372e-02
Sq_pa_CA      2.105457e+00 6.076792e-01  3.4647512 5.307222e-04
Sq_fgat_CA    1.326244e+01 1.443471e+01  0.9187879 3.582065e-01
Sq_rza_CA     5.673575e+00 1.679281e+00  3.3785731 7.286305e-04
Sq_fgat_MA2  -1.735179e-01 6.348847e-02 -2.7330621 6.274851e-03
Sq_rza_MA2   -4.517799e-01 2.048439e-01 -2.2054842 2.742014e-02
Sqrt_fum_CA   2.938854e+01 1.246729e+01  2.3572514 1.841078e-02
Sqrt_top_CA  -2.582823e+00 1.425693e+01 -0.1811626 8.562400e-01
Sqrt_td_CA    6.488121e+02 2.599732e+02  2.4956884 1.257130e-02
Sqrt_rza_CA   3.004242e+01 1.023362e+01  2.9356592 3.328398e-03
Sqrt_pen_MA2  4.581306e-01 1.770250e-01  2.5879431 9.655094e-03

Confusion Matrices:

Train:

p_data   0   1
     0  59  23
     1  50 158

Test:

p_data  0  1
     0 12  6
     1  7  7
```

**Indianapolis Colts**

```
Coefficients:
                Estimate   Std. Error   z value     Pr(>|z|)
(Intercept) -3.598390e+04 1.267026e+04 -2.840029 4.510943e-03
Home.1        9.667154e-01 3.101806e-01  3.116621 1.829364e-03
pa_MA         1.298771e-01 4.847825e-02  2.679079 7.382496e-03
fum_MA       -1.883355e+00 5.806071e-01 -3.243768 1.179600e-03
fum_CA       -1.384712e+02 4.355941e+01 -3.178903 1.478334e-03
pu_CA         1.048749e+02 5.308927e+01  1.975445 4.821769e-02
fgm_CA       -4.371876e+04 1.557738e+04 -2.806555 5.007443e-03
pts_MA2      -1.091354e-01 3.297999e-02 -3.309141 9.358274e-04
pts_CA2       3.066275e+00 9.206231e-01  3.330652 8.664299e-04
pry_CA2      -2.900893e-01 7.163469e-02 -4.049565 5.131302e-05
td_CA2       -2.234860e+01 6.585990e+00 -3.393354 6.904227e-04
sfpy_CA2     -3.926091e-01 1.899110e-01 -2.067332 3.870285e-02
Sq_kry_MA     6.216387e-05 5.640637e-05  1.102072 2.704305e-01
Sq_pu_CA     -1.468557e+01 6.936046e+00 -2.117283 3.423581e-02
Sq_fgm_CA     4.441890e+03 1.595779e+03  2.783524 5.377183e-03
Sq_fgat_CA2 -1.953557e+00 6.795143e-01 -2.874932 4.041150e-03
Sq_sfpy_CA2  5.386083e-04 2.538939e-04  2.121391 3.388890e-02
```

```
Sqrt_pr_MA   1.155682e+00 5.167798e-01  2.236313 2.533126e-02
Sqrt_pts_CA -1.350652e+01 4.647302e+00 -2.906314 3.657145e-03
Sqrt_fum_CA  2.599915e+02 7.666253e+01  3.391376 6.954250e-04
Sqrt_fgm_CA  7.462721e+04 2.648245e+04  2.817987 4.832578e-03
```

Confusion Matrices:

Train:

```
p_data   0   1
     0  63  24
     1  43 160
```

Test:

```
p_data  0  1
     0 19 13
     1  0  1
```

**Kansas City Chiefs**

Coefficients:

```
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -1.050834e+05 3.179127e+04 -3.305416 9.483553e-04
pu_MA         7.377092e-01 2.045709e-01  3.606130 3.107971e-04
sfpy_MA      -4.174742e+00 1.723811e+00 -2.421810 1.544344e-02
ry_CA         2.684553e-01 8.445761e-02  3.178580 1.479983e-03
ra_CA        -3.055962e+00 6.392827e-01 -4.780299 1.750350e-06
py_CA        -1.463531e+00 8.260037e-01 -1.771822 7.642416e-02
pa_CA        -6.123738e+03 1.856125e+03 -3.299206 9.695858e-04
ints_CA      -1.254572e+02 3.720524e+01 -3.372030 7.461644e-04
npy_CA       -2.849073e+00 6.440449e-01 -4.423718 9.701669e-06
ry_MA2       -1.444202e+00 5.778653e-01 -2.499202 1.244735e-02
Sq_sfpy_MA    1.996204e-03 8.089700e-04  2.467587 1.360273e-02
Sq_py_CA      3.025461e-03 1.694258e-03  1.785715 7.414546e-02
Sq_pa_CA      2.974546e+01 9.030213e+00  3.293993 9.877483e-04
Sq_ints_CA    6.338912e+01 1.801917e+01  3.517872 4.350225e-04
Sq_npy_CA     8.390867e-03 1.872446e-03  4.481234 7.421286e-06
Sq_ry_MA2     2.209960e-03 8.375800e-04  2.638506 8.327229e-03
Sqrt_sfpy_MA  1.032094e+02 4.313668e+01  2.392613 1.672887e-02
Sqrt_pa_CA    4.778829e+04 1.447823e+04  3.300699 9.644420e-04
Sqrt_ry_MA2   1.943059e+01 8.134018e+00  2.388806 1.690320e-02
Sqrt_ra_MA2  -6.381652e-01 6.314589e-01 -1.010620 3.121981e-01
Sqrt_pc_MA2  -1.011249e+00 4.310790e-01 -2.345856 1.898345e-02
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 111  42
     1  33  88
```

Test:

```
p_data  0  1
     0  3  3
     1  9 20
```

**Seattle Seahawks**

```
Coefficients:
                  Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)  -1.364752e+03 9.786849e+02 -1.3944757 1.631740e-01
Home.1        1.498690e+00 2.958892e-01  5.0650370 4.083210e-07
ra_MA        -8.826661e-01 4.366933e-01 -2.0212496 4.325393e-02
pa_MA         6.186528e-02 4.556408e-02  1.3577645 1.745384e-01
kry_CA       -2.232470e+01 1.562216e+01 -1.4290410 1.529924e-01
top_CA       -7.543524e-01 7.896303e-01 -0.9553235 3.394141e-01
sfpy_CA       2.209049e+00 1.575835e+00  1.4018274 1.609668e-01
sk_MA2        4.782174e-01 1.807509e-01  2.6457266 8.151566e-03
td_MA2        2.936099e+01 1.703148e+01  1.7239251 8.472137e-02
sfpy_MA2     -7.233186e-02 3.341825e-02 -2.1644420 3.043044e-02
pry_CA2      -1.860984e+00 8.088124e-01 -2.3008846 2.139815e-02
kr_CA2        1.772851e+01 1.234567e+01  1.4360112 1.509992e-01
Sq_ra_MA      1.727929e-02 7.665378e-03  2.2541988 2.418366e-02
Sq_kry_CA     3.659079e-02 2.600792e-02  1.4069097 1.594542e-01
Sq_sfpy_CA   -3.302028e-03 2.242214e-03 -1.4726641 1.408416e-01
Sq_td_MA2    -2.149597e+00 1.185920e+00 -1.8125994 6.989364e-02
Sq_sfpy_MA2   9.987085e-05 4.647829e-05  2.1487636 3.165314e-02
Sqrt_kry_CA   2.968028e+02 2.075536e+02  1.4300060 1.527153e-01
Sqrt_td_MA2  -5.817131e+01 3.442596e+01 -1.6897513 9.107554e-02
Sqrt_pry_CA2  1.724734e+01 7.458381e+00  2.3124778 2.075137e-02
Sqrt_kr_CA2  -7.092901e+01 4.987456e+01 -1.4221481 1.549833e-01

Confusion Matrices:

Train:

p_data   0   1
     0  83  33
     1  42 131

Test:

p_data  0  1
     0  2  3
     1 12 16
```

**Miami Dolphins**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   7.927035e+05 3.372202e+05  2.350700 1.873812e-02
kr_MA         2.302692e+00 9.879477e-01  2.330783 1.976479e-02
top_CA        5.459567e+04 2.243078e+04  2.433962 1.493455e-02
sfpy_CA      -4.927348e+00 1.522507e+00 -3.236338 1.210740e-03
npy_CA       -4.476721e+02 1.658818e+02 -2.698742 6.960215e-03
pr_MA2        3.304920e+00 1.784836e+00  1.851665 6.407388e-02
sk_CA2        3.678997e-01 3.135104e-01  1.173485 2.406014e-01
fgat_CA2      4.416924e+01 1.934162e+01  2.283637 2.239288e-02
Sq_kr_MA     -3.433921e-01 1.428767e-01 -2.403416 1.624268e-02
Sq_rza_MA     1.167558e-01 3.575709e-02  3.265248 1.093684e-03
Sq_fum_CA     2.082541e+01 4.586315e+00  4.540772 5.604867e-06
Sq_kr_CA      1.095783e+00 2.736421e-01  4.004438 6.216522e-05
Sq_top_CA    -2.969403e+02 1.225372e+02 -2.423267 1.538161e-02
Sq_sfpy_CA    6.263577e-03 2.028668e-03  3.087532 2.018258e-03
Sq_npy_CA     3.737790e-01 1.388568e-01  2.691831 7.106088e-03
Sq_fgat_CA2  -1.176555e+01 5.159108e+00 -2.280540 2.257571e-02
Sqrt_top_CA  -4.029377e+05 1.651822e+05 -2.439352 1.471361e-02
```

```
Sqrt_rza_CA  -1.419501e+01 3.255536e+00 -4.360268 1.299033e-05
Sqrt_npy_CA   8.425217e+03 3.118070e+03  2.702062 6.891097e-03
Sqrt_fum_MA2  2.046416e+00 6.059647e-01  3.377121 7.324881e-04
Sqrt_pr_MA2  -1.026074e+01 5.205893e+00 -1.970986 4.872550e-02
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 111  46
     1  33  81
```

Test:

```
p_data  0  1
     0 16 12
     1  2  1
```

**Chicago Bears**

Coefficients:

```
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -3.459443e+04 1.204306e+04 -2.872562 4.071579e-03
kry_CA       -1.068090e+02 6.743843e+01 -1.583800 1.132393e-01
top_CA       -5.496604e+02 2.685266e+02 -2.046950 4.066302e-02
fgat_CA      -1.515548e+04 9.515959e+03 -1.592638 1.112415e-01
sfpy_CA       5.005769e-02 3.143518e-02  1.592410 1.112927e-01
pa_MA2        2.437118e-01 6.664489e-02  3.656872 2.553122e-04
pc_MA2       -3.459754e-01 9.051386e-02 -3.822347 1.321873e-04
pr_CA2       -4.890251e+02 2.152519e+02 -2.271874 2.309414e-02
Sq_ints_CA    9.551154e+00 2.589929e+00  3.687805 2.261965e-04
Sq_kry_CA     2.013222e-01 1.193988e-01  1.686132 9.177031e-02
Sq_td_CA     -4.008128e+00 1.177450e+00 -3.404075 6.638858e-04
Sq_fgat_CA    1.431314e+03 8.908024e+02  1.606769 1.081050e-01
Sq_rzc_CA     4.419205e+00 1.820950e+00  2.426868 1.522980e-02
Sq_sk_MA2     9.366111e-02 3.296671e-02  2.841081 4.496085e-03
Sq_pr_CA2     3.839405e+01 1.703502e+01  2.253830 2.420685e-02
Sqrt_kr_MA   -5.486932e+00 1.323821e+00 -4.144769 3.401559e-05
Sqrt_kry_MA   8.494290e-01 2.283948e-01  3.719125 1.999139e-04
Sqrt_kry_CA   1.334098e+03 8.724498e+02  1.529140 1.262298e-01
Sqrt_top_CA   5.946964e+03 2.894519e+03  2.054560 3.992151e-02
Sqrt_fgat_CA  2.679860e+04 1.691742e+04  1.584083 1.131748e-01
Sqrt_pr_CA2   9.400908e+02 4.144851e+02  2.268093 2.332353e-02
```

Confusion Matrices:

Train:

```
p_data   0   1
     0 102  38
     1  36  95
```

Test:

```
p_data  0  1
     0  8  4
     1  8 13
```

**Minnesota Vikings**

```
Coefficients:
                 Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)   1.719829e+04 5.042291e+03  3.4108095 6.477032e-04
Home.1        1.642190e+00 3.090199e-01  5.3141885 1.071337e-07
pc_CA        -1.102429e+02 3.664650e+01 -3.0082787 2.627320e-03
kry_CA        1.139034e+01 3.930244e+00  2.8981258 3.754000e-03
top_CA        4.085470e+02 1.495720e+02  2.7314397 6.305827e-03
fgat_CA       2.178480e+02 6.097724e+01  3.5726113 3.534392e-04
sfpy_CA      -8.762784e+00 3.760029e+00 -2.3305097 1.977923e-02
pen_MA2       2.741077e+00 1.272894e+00  2.1534207 3.128563e-02
pa_CA2        2.890600e+02 1.560634e+02  1.8521966 6.399760e-02
pen_CA2       9.508719e-02 3.517509e-02  2.7032535 6.866435e-03
Sq_fgat_CA   -6.347071e+01 1.852540e+01 -3.4261445 6.122146e-04
Sq_sfpy_CA    1.246504e-02 5.472570e-03  2.2777298 2.274268e-02
Sq_py_MA2     2.076889e-06 8.907594e-06  0.2331593 8.156377e-01
Sq_pen_MA2   -9.387594e-03 4.174822e-03 -2.2486215 2.453659e-02
Sq_td_MA2    -9.224310e-02 4.818269e-02 -1.9144448 5.556334e-02
Sq_pa_CA2    -1.510407e+00 8.014697e-01 -1.8845466 5.949108e-02
Sqrt_pc_CA    9.777462e+02 3.289778e+02  2.9720734 2.957960e-03
Sqrt_kry_CA  -2.117384e+02 7.534061e+01 -2.8104146 4.947772e-03
Sqrt_top_CA  -4.497843e+03 1.646793e+03 -2.7312744 6.308991e-03
Sqrt_pen_MA2 -2.454975e+01 1.190549e+01 -2.0620532 3.920267e-02
Sqrt_pa_CA2  -2.174197e+03 1.184366e+03 -1.8357466 6.639514e-02

Confusion Matrices:

Train:

p_data   0   1
     0 104  36
     1  37  97

Test:

p_data  0  1
     0 11 13
     1  1  9
```

**Tampa Bay Buccaneers**

```
Coefficients:
                Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  8.670450e+03 2.590025e+03  3.347631 8.150534e-04
pts_CA       1.952195e+00 1.173615e+00  1.663403 9.623180e-02
pc_CA        7.077709e+02 2.819026e+02  2.510693 1.204943e-02
top_CA      -8.448826e+00 2.017275e+00 -4.188238 2.811286e-05
td_CA       -1.738630e+01 7.943860e+00 -2.188646 2.862258e-02
rzc_CA      -1.016184e+01 4.331595e+00 -2.345980 1.897712e-02
npy_CA       8.828733e+00 4.154464e+00  2.125120 3.357664e-02
rza_MA2      7.672146e-01 2.738250e-01  2.801843 5.081167e-03
ra_CA2      -6.941472e+00 3.222182e+00 -2.154277 3.121846e-02
Sq_rza_MA    1.883860e-01 4.578021e-02  4.115010 3.871635e-05
Sq_py_CA     8.435022e-04 2.601897e-04  3.241874 1.187466e-03
Sq_pc_CA    -6.203828e+00 2.636664e+00 -2.352908 1.862724e-02
Sq_kr_CA     1.536290e+01 6.157593e+00  2.494953 1.259739e-02
Sq_pts_MA2  -2.974804e-03 9.300759e-04 -3.198454 1.381668e-03
Sq_pen_CA2   6.432241e-04 2.891403e-04  2.224609 2.610748e-02
Sqrt_pen_MA  5.743616e-01 1.915316e-01  2.998782 2.710608e-03
Sqrt_pc_CA  -4.086347e+03 1.581586e+03 -2.583702 9.774620e-03
```

```
Sqrt_pu_CA   1.575260e+02 3.958963e+01  3.978972 6.921389e-05
Sqrt_kr_CA  -3.527852e+02 1.462553e+02 -2.412120 1.586008e-02
Sqrt_npy_CA -2.610107e+02 1.140001e+02 -2.289566 2.204647e-02
Sqrt_ra_CA2  7.250413e+01 3.382465e+01  2.143530 3.207059e-02

Confusion Matrices:

Train:

p_data   0   1
     0 119  46
     1  32  75

Test:

p_data  0  1
     0  3  3
     1 19  7
```

**New England Patriots**

```
Coefficients:
                  Estimate   Std. Error   z value    Pr(>|z|)
(Intercept)   2.107935e+04 7.874893e+03  2.676779 0.007433366
Home.1        9.118794e-01 3.116019e-01  2.926425 0.003428826
ry_CA        -1.174757e+02 4.593522e+01 -2.557420 0.010545183
py_CA         6.027860e+00 2.242823e+00  2.687621 0.007196293
pu_CA        -2.219707e+02 7.241170e+01 -3.065398 0.002173803
kry_CA        2.936684e-01 1.197304e-01  2.452746 0.014177047
sfpy_CA      -9.870478e+00 3.379868e+00 -2.920374 0.003496121
ra_CA2       -1.028876e+02 6.855585e+01 -1.500785 0.133411097
rza_CA2      -3.010825e+00 1.594988e+00 -1.887678 0.059069145
Sq_sfpy_CA    1.267665e-02 4.385205e-03  2.890777 0.003842910
Sq_pts_CA2   -5.258185e-03 2.155205e-03 -2.439761 0.014696979
Sq_ra_CA2     5.987446e-01 4.079698e-01  1.467620 0.142207553
Sqrt_pts_MA   4.217278e-01 3.385355e-01  1.245742 0.212859237
Sqrt_pa_MA    1.361107e+00 6.334669e-01  2.148663 0.031661112
Sqrt_ry_CA    4.810449e+03 1.845592e+03  2.606453 0.009148548
Sqrt_ra_CA    1.627850e+01 1.228797e+01  1.324751 0.185253878
Sqrt_py_CA   -1.849338e+02 6.651274e+01 -2.780426 0.005428755
Sqrt_pu_CA    9.743806e+02 3.151182e+02  3.092111 0.001987385
Sqrt_ra_CA2   7.311284e+02 4.831814e+02  1.513155 0.130240237
Sqrt_rza_CA2  1.201269e+01 4.091755e+00  2.935828 0.003326582
log_ry_CA    -1.232445e+04 4.631304e+03 -2.661119 0.007788149

Confusion Matrices:

Train:

p_data   0   1
     0  29   6
     1  47 218

Test:

p_data  0  1
     0  0  0
     1  9 29
```

**Detroit Lions**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   2.983847e+04 8.724549e+03  3.420059 6.260763e-04
pc_MA         5.283601e-01 1.415880e-01  3.731674 1.902112e-04
ints_MA       3.373877e+00 2.188389e+00  1.541717 1.231423e-01
pu_MA        -1.000494e+01 3.822561e+00 -2.617339 8.861832e-03
pry_MA        6.003860e-02 1.911223e-02  3.141372 1.681585e-03
pts_CA        1.209541e+02 2.789144e+01  4.336603 1.447016e-05
ry_CA         5.768270e+02 1.742222e+02  3.310870 9.300652e-04
pa_CA         1.271149e+02 4.546240e+01  2.796044 5.173242e-03
rzc_CA       -2.137406e+01 6.651826e+00 -3.213262 1.312366e-03
rzc_MA2      -1.192630e+00 2.846267e-01 -4.190155 2.787634e-05
Sq_py_MA      5.043813e-05 2.428369e-05  2.077037 3.779814e-02
Sq_pa_MA     -7.322612e-03 1.647369e-03 -4.445034 8.787816e-06
Sq_td_MA     -2.700387e-01 9.428503e-02 -2.864068 4.182386e-03
Sq_pts_CA    -3.164613e+00 7.371084e-01 -4.293280 1.760525e-05
Sq_ry_CA     -1.042653e+00 3.122226e-01 -3.339453 8.394369e-04
Sqrt_ints_MA -6.455960e+00 4.571815e+00 -1.412122 1.579141e-01
Sqrt_pu_MA    4.531553e+01 1.672622e+01  2.709251 6.743533e-03
Sqrt_ry_CA   -7.376383e+03 2.239696e+03 -3.293475 9.895711e-04
Sqrt_py_CA   -8.610193e+00 3.293228e+00 -2.614514 8.935439e-03
Sqrt_pa_CA   -1.455389e+03 5.298428e+02 -2.746831 6.017413e-03
Sqrt_sfpy_CA -1.187910e+01 3.661422e+00 -3.244394 1.177009e-03

Confusion Matrices:

Train:

p_data   0   1
     0 155  41
     1  20  51

Test:

p_data  0  1
     0 14 11
     1  3  4
```

**New Orleans Saints**

```
Coefficients:
                  Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)   2.225088e+04 1.958023e+04  1.1363951 2.557912e-01
fgat_MA       3.665816e+00 1.590596e+00  2.3046804 2.118448e-02
pts_CA       -1.486695e+02 7.942876e+02 -0.1871734 8.515247e-01
pu_CA        -1.707697e+04 6.128744e+03 -2.7863735 5.330141e-03
pen_CA       -4.090406e-01 1.556799e-01 -2.6274466 8.602834e-03
fgm_CA        1.692289e+02 7.550403e+01  2.2413222 2.500521e-02
rzc_CA       -5.788069e+00 1.564455e+00 -3.6997353 2.158244e-04
sfpy_CA       3.424708e+02 9.936910e+01  3.4464518 5.680000e-04
fgat_CA2      3.603006e+01 1.713696e+01  2.1024767 3.551154e-02
rza_CA2       1.122072e+00 5.963837e-01  1.8814599 5.990938e-02
Sq_fgat_MA   -1.037292e+00 4.320856e-01 -2.4006632 1.636539e-02
Sq_npy_MA     3.624952e-05 1.699952e-05  2.1323845 3.297525e-02
Sq_pts_CA     1.158936e+00 5.699915e+00  0.2033252 8.388809e-01
Sq_ints_CA    1.124072e+01 2.551584e+00  4.4053881 1.055947e-05
Sq_pu_CA      6.242235e+02 2.234465e+02  2.7936152 5.212247e-03
Sq_sfpy_CA   -1.565088e-01 4.536217e-02 -3.4502050 5.601611e-04
Sq_fgat_CA2  -9.951835e+00 4.704077e+00 -2.1155765 3.438084e-02
```

```
Sqrt_pts_CA   9.083459e+02 5.100246e+03  0.1780984 8.586457e-01
Sqrt_pu_CA    4.854794e+04 1.746292e+04  2.7800583 5.434914e-03
Sqrt_fgm_CA  -3.943908e+02 1.818987e+02 -2.1681899 3.014424e-02
Sqrt_sfpy_CA -8.716077e+03 2.530224e+03 -3.4447843 5.715153e-04

Confusion Matrices:

Train:

p_data   0   1
     0  74  38
     1  53 110

Test:

p_data  0  1
     0 10 26
     1  0  0
```

**Arizona Cardinals**

```
Coefficients:
                   Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)    2.423468e+02 9.533461e+01  2.542065 0.0110199677
pen_MA         1.634740e-03 1.199845e-02  0.136246 0.8916268045
ints_CA        1.047171e+01 4.277000e+00  2.448377 0.0143501474
pry_CA        -2.987688e+00 2.372819e+00 -1.259130 0.2079833608
kry_CA        -4.384492e-01 1.472928e-01 -2.976719 0.0029135111
top_CA        -1.643553e+00 6.949463e-01 -2.365007 0.0180297335
pts_MA2        7.417148e-01 3.599468e-01  2.060624 0.0393389623
pts_CA2        2.936647e+00 8.890186e-01  3.303246 0.0009557238
ints_CA2       4.048956e+02 1.686921e+02  2.400205 0.0163858818
pr_CA2         2.067334e+01 9.385291e+00  2.202738 0.0276132065
fgm_CA2       -2.033446e+01 1.302779e+01 -1.560853 0.1185585518
Sq_npy_MA     -2.923617e-05 8.878682e-06 -3.292850 0.0009917729
Sq_pry_CA      8.838754e-02 6.366125e-02  1.388404 0.1650140386
Sq_pts_CA2    -6.726021e-02 2.025635e-02 -3.320450 0.0008987242
Sq_ints_CA2   -5.718960e+01 2.600376e+01 -2.199282 0.0278578757
Sq_pr_CA2     -4.566175e+00 2.065938e+00 -2.210219 0.0270899787
Sq_fgm_CA2     6.494785e+00 4.352339e+00  1.492252 0.1356332147
Sqrt_kr_CA     3.676543e+01 1.325827e+01  2.773019 0.0055538883
Sqrt_pts_MA2  -6.914285e+00 3.287763e+00 -2.103036 0.0354626094
Sqrt_ry_CA2   -6.884529e-01 3.596796e-01 -1.914073 0.0556108769
Sqrt_ints_CA2 -5.777215e+02 2.326328e+02 -2.483405 0.0130132866

Confusion Matrices:

Train:

p_data   0   1
     0 111  46
     1  41  77

Test:

p_data  0  1
     0 12  9
     1  9  2
```

**New York Giants**

```
Coefficients:
                   Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)    4.891866e+04 1.186145e+04  4.124172 3.720700e-05
fgat_MA        3.375386e+00 6.963141e-01  4.847505 1.250241e-06
ry_CA          4.281439e+02 1.456260e+02  2.940024 3.281863e-03
ra_CA         -5.010677e+01 1.350501e+01 -3.710235 2.070667e-04
py_CA          8.278232e+00 3.055266e+00  2.709496 6.738550e-03
pry_CA        -1.292145e+01 7.976980e+00 -1.619843 1.052661e-01
pen_CA        -3.849732e-01 1.626247e-01 -2.367249 1.792088e-02
td_CA          1.816787e+04 9.015542e+03  2.015172 4.388661e-02
Sq_pry_MA     -1.587090e-03 4.966818e-04 -3.195385 1.396445e-03
Sq_rzc_MA     -2.668883e-01 1.118617e-01 -2.385878 1.703839e-02
Sq_ry_CA      -5.802926e-01 1.948994e-01 -2.977395 2.907094e-03
Sq_ra_CA       7.689643e-01 2.195679e-01  3.502171 4.614829e-04
Sq_td_CA      -1.349721e+03 6.504737e+02 -2.074982 3.798821e-02
Sq_ints_MA2    4.305354e-01 1.532594e-01  2.809195 4.966562e-03
Sqrt_pts_MA    1.536298e+00 5.400339e-01  2.844819 4.443673e-03
Sqrt_fgm_MA   -8.129907e+00 1.787802e+00 -4.547431 5.430461e-06
Sqrt_ry_CA    -6.320502e+03 2.164376e+03 -2.920242 3.497595e-03
Sqrt_py_CA    -2.402382e+02 8.959752e+01 -2.681304 7.333593e-03
Sqrt_pry_CA    1.261629e+02 7.102875e+01  1.776223 7.569616e-02
Sqrt_td_CA    -3.626860e+04 1.825734e+04 -1.986522 4.697539e-02
Sqrt_fgat_CA  -6.069575e+01 1.442310e+01 -4.208233 2.573759e-05

Confusion Matrices:

Train:

p_data   0   1
     0  83  40
     1  47 112

Test:

p_data  0  1
     0  0  0
     1 24  8
```

**San Diego/Los Angeles Charges**

```
Coefficients:
                   Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)    5.491935e+05 2.111470e+05  2.601001 9.295229e-03
pts_CA        -4.287746e+01 1.581643e+01 -2.710944 6.709198e-03
ry_CA          4.308051e+00 1.949889e+00  2.209383 2.714803e-02
sk_CA          9.414301e+01 3.198035e+01  2.943777 3.242340e-03
pr_CA         -3.233419e+02 9.633967e+01 -3.356270 7.900138e-04
pen_CA        -1.718407e+01 5.459991e+00 -3.147271 1.648023e-03
top_CA         3.756380e+04 1.417392e+04  2.650206 8.044265e-03
sfpy_CA       -2.226085e+01 5.824514e+00 -3.821923 1.324149e-04
pry_CA2       -2.289384e-01 5.111826e-02 -4.478604 7.513294e-06
Sq_ints_MA    -4.776299e-01 1.828295e-01 -2.612433 8.990031e-03
Sq_sk_CA      -2.061190e+01 6.970816e+00 -2.956885 3.107640e-03
Sq_pr_CA       6.488960e+01 1.889053e+01  3.435033 5.924811e-04
Sq_pen_CA      1.578387e-01 4.847642e-02  3.255989 1.129980e-03
Sq_top_CA     -2.110790e+02 7.909705e+01 -2.668607 7.616643e-03
Sq_rza_CA2    -2.323957e-01 1.114463e-01 -2.085271 3.704470e-02
Sqrt_pts_CA    3.893046e+02 1.441030e+02  2.701573 6.901243e-03
Sqrt_ry_CA    -8.352332e+01 3.971163e+01 -2.103246 3.544430e-02
```

```
Sqrt_top_CA  -2.727547e+05 1.032790e+05 -2.640949 8.267404e-03
Sqrt_fgm_CA   4.813511e+01 1.880120e+01  2.560215 1.046074e-02
Sqrt_sfpy_CA  8.459552e+02 2.233445e+02  3.787670 1.520668e-04
Sqrt_fum_MA2  2.000969e+00 6.112080e-01  3.273793 1.061141e-03
```

```
Confusion Matrices:

Train:

p_data   0   1
     0 100  28
     1  37 110

Test:

p_data  0  1
     0 12 22
     1  0  0
```

**Oakland Raiders**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)    6.042588e+02 3.257462e+02  1.854999 6.359638e-02
ra_CA         -1.448229e+01 5.784126e+00 -2.503800 1.228675e-02
pa_CA         -4.407720e+00 1.331337e+00 -3.310747 9.304730e-04
pc_CA          8.466341e+00 1.874801e+00  4.515862 6.305970e-06
pry_CA         5.633453e+00 2.245937e+00  2.508286 1.213184e-02
rzc_CA        -1.463609e+01 3.112642e+00 -4.702144 2.574437e-06
pry_MA2       -2.193309e-01 8.248014e-02 -2.659196 7.832735e-03
sk_CA2         9.233461e-01 4.454424e-01  2.072874 3.818400e-02
fgat_CA2       6.541291e+02 3.376632e+02  1.937223 5.271807e-02
Sq_ra_CA       2.882367e-01 1.031377e-01  2.794679 5.195121e-03
Sq_pry_CA     -1.507737e-01 5.820495e-02 -2.590392 9.586658e-03
Sq_kr_CA       1.184289e+00 3.999780e-01  2.960885 3.067568e-03
Sq_sfpy_CA    -3.296771e-04 8.142635e-05 -4.048776 5.148612e-05
Sq_pts_CA2    -4.689551e-02 1.714297e-02 -2.735553 6.227547e-03
Sq_pry_CA2    -4.213830e-03 1.291273e-03 -3.263315 1.101171e-03
Sq_kry_CA2    -2.903587e-04 1.101717e-04 -2.635511 8.401074e-03
Sq_td_CA2      2.833274e+00 1.063145e+00  2.664993 7.698995e-03
Sq_fgat_CA2   -5.731889e+01 3.054781e+01 -1.876367 6.060495e-02
Sqrt_pu_CA     6.205035e+01 1.854018e+01  3.346804 8.174901e-04
Sqrt_pry_MA2   2.225988e+00 7.816271e-01  2.847890 4.401018e-03
Sqrt_fgat_CA2 -1.181470e+03 6.093632e+02 -1.938860 5.251836e-02
```

```
Confusion Matrices:

Train:

p_data   0   1
     0 145  45
     1  22  63

Test:

p_data  0  1
     0 18  8
     1  4  2
```

**Baltimore Ravens**

```
Coefficients:
                  Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)   2.541975e+02 1.294581e+02  1.9635502 4.958228e-02
Home.1        1.606568e+00 3.079412e-01  5.2171258 1.817209e-07
kr_MA         1.599366e+00 9.079025e-01  1.7616053 7.813601e-02
kry_MA       -6.884891e-03 1.513318e-02 -0.4549534 6.491428e-01
pry_CA       -1.794127e-01 1.598919e-01 -1.1220871 2.618254e-01
pen_CA       -1.576295e+01 4.516415e+00 -3.4901461 4.827565e-04
top_CA2       1.095283e+01 4.740300e+00  2.3105770 2.085623e-02
rza_CA2      -1.866419e+01 5.600870e+00 -3.3323739 8.610850e-04
Sq_kr_MA     -1.868975e-01 1.113138e-01 -1.6790153 9.314906e-02
Sq_sk_CA      2.096597e+00 6.096427e-01  3.4390578 5.837425e-04
Sq_pen_CA     1.531840e-01 4.360625e-02  3.5128906 4.432599e-04
Sq_npy_CA     8.077265e-04 2.600480e-04  3.1060667 1.895940e-03
Sq_td_MA2    -1.148334e-01 4.325870e-02 -2.6545739 7.940864e-03
Sq_kr_CA2     1.719530e-01 7.828185e-02  2.1965886 2.804984e-02
Sq_top_CA2   -1.709774e-01 7.665944e-02 -2.2303504 2.572419e-02
Sq_rza_CA2    2.132448e+00 7.209900e-01  2.9576669 3.099769e-03
Sqrt_pu_CA   -5.207265e+01 1.758654e+01 -2.9609380 3.067037e-03
Sqrt_kr_CA    3.557909e+01 1.293365e+01  2.7508918 5.943327e-03
Sqrt_rza_CA2  1.512654e+01 5.546251e+00  2.7273442 6.384640e-03
log_py_MA     2.729835e+00 1.273408e+00  2.1437242 3.205499e-02
log_pa_MA    -4.494590e+00 1.473444e+00 -3.0503971 2.285390e-03

Confusion Matrices:

Train:

p_data   0   1
     0  79  35
     1  43 132

Test:

p_data  0  1
     0  5  6
     1  9 13
```

**Pittsburgh Steelers**

```
Coefficients:
                  Estimate   Std. Error     z value     Pr(>|z|)
(Intercept)   42.085799784 2.599775e+02  0.16188248 8.713984e-01
Home.1         0.908497361 2.954240e-01  3.07523242 2.103386e-03
sk_MA          5.706800692 1.825629e+00  3.12593671 1.772397e-03
npy_MA        -0.018796282 5.239531e-03 -3.58739794 3.339944e-04
sk_CA          3.765838964 6.557352e+01  0.05742926 9.542033e-01
pu_CA        -78.163391425 3.156141e+01 -2.47654936 1.326593e-02
rza_CA         4.244955900 1.158746e+00  3.66340474 2.488848e-04
sfpy_CA       -2.227890012 9.745864e-01 -2.28598522 2.225512e-02
ra_MA2         0.522872149 3.493363e-01  1.49675859 1.344561e-01
sk_MA2        -3.296941851 1.193312e+00 -2.76284916 5.729924e-03
pry_CA2        0.888414790 4.045319e-01  2.19615535 2.808083e-02
rza_CA2       -2.459006975 6.241258e-01 -3.93992174 8.150819e-05
Sq_pts_CA     -0.081280496 3.686090e-02 -2.20506011 2.744988e-02
Sq_td_CA       3.846207178 2.031278e+00  1.89349145 5.829254e-02
Sq_sfpy_CA     0.002919089 1.301934e-03  2.24211730 2.495379e-02
Sq_ra_MA2     -0.009839669 6.104471e-03 -1.61187918 1.069882e-01
Sqrt_sk_MA   -16.762960453 5.602330e+00 -2.99214071 2.770285e-03
```

```
Sqrt_sk_CA    -9.622333479 2.037087e+02 -0.04723576 9.623253e-01
Sqrt_pu_CA   367.517411585 1.513825e+02  2.42774109 1.519319e-02
Sqrt_sk_MA2    9.809951736 3.526110e+00  2.78208909 5.401021e-03
Sqrt_pry_CA2 -10.191660236 4.037305e+00 -2.52437231 1.159051e-02

Confusion Matrices:

Train:

p_data   0   1
     0  53  21
     1  49 167

Test:

p_data  0  1
     0  6  8
     1  5 14
```

**Carolina Panthers**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -5.773782e+04 1.641185e+04 -3.518058 4.347177e-04
kr_MA        -3.018250e+00 9.536083e-01 -3.165083 1.550386e-03
rzc_MA        4.558140e+00 1.331484e+00  3.423354 6.185355e-04
ry_CA        -4.216211e+00 1.034750e+00 -4.074619 4.608979e-05
py_CA        -1.115643e-01 5.318512e-02 -2.097659 3.593527e-02
rza_CA        5.163555e+00 1.390147e+00  3.714396 2.036896e-04
sfpy_CA      -3.152483e+02 8.978983e+01 -3.510958 4.464950e-04
ints_MA2      7.506320e-01 3.116891e-01  2.408272 1.602825e-02
pu_CA2       -1.306937e+03 5.088951e+02 -2.568186 1.022323e-02
npy_CA2       3.253347e+01 1.183034e+01  2.750003 5.959471e-03
Sq_kr_MA      4.271847e-01 1.397254e-01  3.057315 2.233292e-03
Sq_sfpy_CA    1.446516e-01 4.088745e-02  3.537799 4.034776e-04
Sq_sk_MA2     8.543869e-02 3.650613e-02  2.340393 1.926346e-02
Sq_pu_CA2     4.711615e+01 1.791820e+01  2.629513 8.550719e-03
Sq_pry_CA2   -3.948100e-03 1.175464e-03 -3.358758 7.829371e-04
Sq_npy_CA2   -3.275812e-02 1.166178e-02 -2.809016 4.969317e-03
Sqrt_rzc_MA  -1.073592e+01 3.211530e+00 -3.342929 8.289921e-04
Sqrt_ry_CA    7.895540e+01 2.016334e+01  3.915790 9.010867e-05
Sqrt_sfpy_CA  8.001007e+03 2.288745e+03  3.495806 4.726322e-04
Sqrt_pu_CA2   3.736895e+03 1.472903e+03  2.537096 1.117765e-02
Sqrt_npy_CA2 -5.562088e+02 2.045160e+02 -2.719634 6.535411e-03

Confusion Matrices:

Train:

p_data   0   1
     0 105  32
     1  35 109

Test:

p_data  0  1
     0  6 11
     1  9  7
```

**Washington Redskins**

```
Coefficients:
                 Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -8.119719e+03 5.490259e+03 -1.478932 0.1391585603
rzc_MA        7.188100e+00 2.007012e+00  3.581493 0.0003416369
pts_CA       -5.783479e+03 2.011051e+03 -2.875849 0.0040294200
pr_CA         3.650152e+03 2.423868e+03  1.505921 0.1320875433
top_CA       -2.035938e+02 1.058476e+02 -1.923461 0.0544222005
td_CA         3.593541e+04 1.384567e+04  2.595427 0.0094473599
pry_MA2       2.666607e-01 9.293066e-02  2.869459 0.0041117439
top_MA2      -3.682035e+00 1.077705e+00 -3.416553 0.0006341930
Sq_rzc_MA    -2.244131e+00 6.141708e-01 -3.653920 0.0002582673
Sq_pts_CA     5.488989e+01 1.910841e+01  2.872551 0.0040717196
Sq_pr_CA     -2.208150e+02 1.602942e+02 -1.377561 0.1683387906
Sq_top_CA     3.310553e+00 1.729504e+00  1.914163 0.0555992970
Sq_td_CA     -3.117272e+03 1.202968e+03 -2.591316 0.0095609559
Sq_top_MA2    6.287819e-02 1.796387e-02  3.500259 0.0004648061
Sqrt_pts_CA   3.227644e+04 1.122669e+04  2.874973 0.0040406245
Sqrt_pr_CA   -7.976430e+03 5.120247e+03 -1.557821 0.1192756320
Sqrt_pry_CA  -4.271919e+00 1.814537e+00 -2.354275 0.0185588519
Sqrt_td_CA   -6.632190e+04 2.555832e+04 -2.594924 0.0094611888
Sqrt_ry_MA2  -4.627500e-01 1.503816e-01 -3.077173 0.0020897429
Sqrt_pry_MA2 -2.446458e+00 8.426352e-01 -2.903342 0.0036920309
Sqrt_top_CA2 -5.545396e+00 1.983371e+00 -2.795945 0.0051748216

Confusion Matrices:

Train:

p_data   0   1
     0 133  52
     1  25  62

Test:

p_data  0  1
     0  4  4
     1 13 10
```

**Tennessee Titans**

```
Coefficients:
                 Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)  -8.232232e+04 3.189576e+04 -2.580980 9.852036e-03
ry_CA        -9.738242e+02 4.493457e+02 -2.167205 3.021921e-02
kry_CA        1.850567e-01 7.200199e-02  2.570160 1.016515e-02
sfpy_CA      -1.192363e+02 6.889715e+01 -1.730642 8.351569e-02
pts_CA2       8.345559e+00 4.404324e+00  1.894856 5.811152e-02
td_CA2       -5.463182e+01 3.074322e+01 -1.777037 7.556219e-02
rza_CA2      -4.001323e+00 9.741480e-01 -4.107510 3.999469e-05
Sq_pa_MA      3.956264e-03 1.157835e-03  3.416949 6.332705e-04
Sq_ry_CA      1.328460e+00 6.159560e-01  2.156745 3.102557e-02
Sq_sfpy_CA    5.117295e-02 2.972148e-02  1.721750 8.511486e-02
Sq_sk_CA2     2.422447e-01 8.169374e-02  2.965279 3.024084e-03
Sqrt_ra_MA    4.665698e+00 9.965173e-01  4.682004 2.840833e-06
Sqrt_top_MA  -6.281597e+00 1.771359e+00 -3.546202 3.908267e-04
Sqrt_ry_CA    1.434258e+04 6.604573e+03  2.171614 2.988482e-02
Sqrt_pa_CA   -1.169185e+01 5.883722e+00 -1.987151 4.690561e-02
Sqrt_pr_CA   -1.888005e+01 8.288992e+00 -2.277726 2.274292e-02
Sqrt_sfpy_CA  3.128538e+03 1.803893e+03  1.734326 8.286028e-02
```

```
Sqrt_td_MA2  -3.984184e+00 1.113576e+00 -3.577828 3.464613e-04
Sqrt_rzc_MA2  3.214560e+00 1.077750e+00  2.982657 2.857580e-03
Sqrt_pts_CA2 -7.483223e+01 3.884511e+01 -1.926426 5.405122e-02
Sqrt_td_CA2   1.711831e+02 9.189739e+01  1.862763 6.249558e-02

Confusion Matrices:

Train:

p_data   0   1
     0 107  34
     1  36  96

Test:

p_data  0  1
     0 13 16
     1  2  3
```

**Buffalo Bills**

```
Coefficients:
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept)   3.837520e+03 1.936403e+03  1.981778 4.750412e-02
Home.1        8.322395e-01 3.058629e-01  2.720956 6.509341e-03
pts_CA       -1.399563e+01 4.999697e+00 -2.799295 5.121438e-03
pr_CA         3.405271e+03 1.712745e+03  1.988196 4.679007e-02
fgm_CA        3.650933e+01 1.531412e+01  2.384030 1.712422e-02
pu_MA2        1.568565e+02 4.249265e+01  3.691380 2.230405e-04
rza_MA2      -6.335438e-01 2.442222e-01 -2.594129 9.483096e-03
npy_MA2      -2.777018e+00 8.868654e-01 -3.131274 1.740498e-03
Sq_pu_CA     -1.253271e+00 3.541746e-01 -3.538567 4.023049e-04
Sq_pr_CA     -2.512605e+02 1.246825e+02 -2.015202 4.388347e-02
Sq_td_CA      2.142923e+01 7.786969e+00  2.751935 5.924426e-03
Sq_pu_MA2    -5.702319e+00 1.523371e+00 -3.743224 1.816742e-04
Sq_pen_MA2   -3.077965e-04 1.054540e-04 -2.918775 3.514098e-03
Sq_fgat_MA2   2.877995e-01 8.117705e-02  3.545330 3.921211e-04
Sq_npy_MA2    2.776650e-03 8.517726e-04  3.259849 1.114716e-03
Sq_fgat_CA2  -1.428693e+00 3.233377e-01 -4.418579 9.935218e-06
Sqrt_pa_CA    1.300995e+01 4.463232e+00  2.914917 3.557828e-03
Sqrt_pr_CA   -6.792400e+03 3.442774e+03 -1.972944 4.850193e-02
Sqrt_npy_CA   1.248631e+01 3.077701e+00  4.057024 4.970195e-05
Sqrt_pu_MA2  -4.396838e+02 1.210928e+02 -3.630964 2.823641e-04
Sqrt_npy_MA2  4.687706e+01 1.542285e+01  3.039455 2.370067e-03

Confusion Matrices:

Train:

p_data   0   1
     0 132  47
     1  25  60

Test:

p_data  0  1
     0  5  5
     1 13 10
```

**Denver Broncos**

Coefficients:

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 3.408619e+03 | 1.627309e+03 | 2.094635 | 3.620341e-02 |
| ra_MA | -1.323829e-01 | 5.137047e-02 | -2.577023 | 9.965532e-03 |
| pry_MA | -2.251889e-01 | 7.793338e-02 | -2.889504 | 3.858495e-03 |
| rzc_MA | -5.152739e-01 | 3.154516e-01 | -1.633448 | 1.023748e-01 |
| ry_CA | 8.600888e-01 | 1.920728e-01 | 4.477930 | 7.537016e-06 |
| pa_CA | 4.343441e+00 | 9.748341e-01 | 4.455569 | 8.367079e-06 |
| pr_CA | -4.243861e+02 | 1.280248e+02 | -3.314875 | 9.168411e-04 |
| fgat_CA | 2.639293e+03 | 1.357415e+03 | 1.944352 | 5.185296e-02 |
| npy_CA | 4.366137e-01 | 9.914013e-02 | 4.404006 | 1.062702e-05 |
| pry_MA2 | -4.163020e-02 | 1.498788e-02 | -2.777591 | 5.476348e-03 |
| top_MA2 | 2.112792e-01 | 8.175935e-02 | 2.584160 | 9.761654e-03 |
| Sq_pu_MA | -4.471432e-02 | 2.131777e-02 | -2.097513 | 3.594815e-02 |
| Sq_pry_MA | 4.718071e-03 | 1.631235e-03 | 2.892330 | 3.823955e-03 |
| Sq_pr_CA | 9.740242e+01 | 2.927461e+01 | 3.327197 | 8.772422e-04 |
| Sq_kry_CA | 1.601977e-03 | 4.062221e-04 | 3.943600 | 8.026756e-05 |
| Sq_fgat_CA | -1.776899e+02 | 9.729453e+01 | -1.826309 | 6.780366e-02 |
| Sq_pc_MA2 | -3.198599e-03 | 1.327282e-03 | -2.409887 | 1.595747e-02 |
| Sq_ints_MA2 | 5.430596e-01 | 1.628432e-01 | 3.334863 | 8.534154e-04 |
| Sqrt_pts_CA | -1.939439e+01 | 4.112522e+00 | -4.715936 | 2.406016e-06 |
| Sqrt_fgat_CA | -5.480505e+03 | 2.748293e+03 | -1.994149 | 4.613576e-02 |
| Sqrt_ry_MA2 | -3.722849e-01 | 1.600822e-01 | -2.325586 | 2.004064e-02 |

Confusion Matrices:

Train:

| p_data | 0 | 1 |
|---|---|---|
| 0 | 67 | 32 |
| 1 | 44 | 135 |

Test:

| p_data | 0 | 1 |
|---|---|---|
| 0 | 5 | 3 |
| 1 | 16 | 8 |

**St. Louis/Los Angeles Rams**

Coefficients:

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 1.040525e+04 | 3.429205e+03 | 3.034305 | 2.410906e-03 |
| pa_CA | -8.333317e+01 | 3.721109e+01 | -2.239471 | 2.512527e-02 |
| pr_CA | 7.307056e+03 | 2.795608e+03 | 2.613763 | 8.955125e-03 |
| kr_CA | 1.408031e+02 | 3.869817e+01 | 3.638495 | 2.742359e-04 |
| top_CA | -3.746284e+00 | 9.426766e-01 | -3.974093 | 7.064803e-05 |
| pc_MA2 | -1.996639e+01 | 8.410122e+00 | -2.374090 | 1.759225e-02 |
| kr_MA2 | 4.450557e-01 | 1.951686e-01 | 2.280365 | 2.258603e-02 |
| top_MA2 | 2.547537e-01 | 8.699526e-02 | 2.928363 | 3.407522e-03 |
| fgat_MA2 | -1.135061e+00 | 3.266790e-01 | -3.474545 | 5.117213e-04 |
| kr_CA2 | 6.219783e+02 | 2.610582e+02 | 2.382527 | 1.719425e-02 |
| pen_CA2 | -1.307764e+00 | 4.755325e-01 | -2.750104 | 5.957637e-03 |
| Sq_pa_CA | 1.168229e+00 | 5.151215e-01 | 2.267871 | 2.333707e-02 |
| Sq_pr_CA | -5.516568e+02 | 2.182490e+02 | -2.527648 | 1.148293e-02 |
| Sq_kr_CA | -1.589608e+01 | 4.298807e+00 | -3.697788 | 2.174861e-04 |
| Sq_pc_MA2 | 1.584225e-01 | 6.811328e-02 | 2.325867 | 2.002563e-02 |
| Sq_kr_CA2 | -2.482450e+01 | 1.095433e+01 | -2.266181 | 2.344029e-02 |

```
Sq_pen_CA2    1.289988e-02 4.655544e-03  2.770864 5.590771e-03
Sqrt_pr_CA   -1.443516e+04 5.438295e+03 -2.654354 7.946041e-03
Sqrt_pc_MA2   1.209974e+02 5.044527e+01  2.398587 1.645846e-02
Sqrt_rzc_MA2 -2.188216e+00 6.771147e-01 -3.231677 1.230661e-03
Sqrt_kr_CA2  -1.688636e+03 6.920103e+02 -2.440189 1.467957e-02

Confusion Matrices:

Train:

p_data   0   1
     0 141  51
     1  22  59

Test:

p_data  0  1
     0  9 23
     1  1  3
```

**Houston Texans**

```
Coefficients:
                 Estimate  Std. Error    z value     Pr(>|z|)
(Intercept) -1.264912e+02 384.5053985 -0.3289711 7.421776e-01
Home.1       1.304969e+00   0.3429893  3.8046926 1.419804e-04
ra_MA        2.295650e+00   1.0482931  2.1898931 2.853199e-02
ints_MA      1.317120e+00   0.4899986  2.6880074 7.187981e-03
kr_MA       -8.690203e-01   0.3139809 -2.7677486 5.644497e-03
sk_CA        5.128120e+02 124.4190784  4.1216509 3.761669e-05
ints_CA     -1.971397e+01   5.7486147 -3.4293427 6.050451e-04
fum_CA       3.874269e+01   9.5429302  4.0598314 4.910817e-05
rza_CA      -1.164422e+03 391.8219755 -2.9718143 2.960457e-03
rzc_CA      -4.684305e+01  10.7603430 -4.3533044 1.341007e-05
ry_CA2      -3.586186e-02   0.0179812 -1.9944084 4.610744e-02
pu_CA2       2.410190e-01   0.5897160  0.4087035 6.827573e-01
rza_CA2     -2.882774e+00   1.0752066 -2.6811347 7.337298e-03
Sq_fum_MA   -1.504889e+00   0.4130284 -3.6435493 2.689041e-04
Sq_sk_CA    -1.850649e+01   4.7430573 -3.9018067 9.547741e-05
Sq_pu_CA    -6.445085e-01   0.2151157 -2.9961016 2.734553e-03
Sq_fgat_CA  -1.007940e+01   2.1273339 -4.7380446 2.157904e-06
Sq_rza_CA    8.063439e+01  28.9018871  2.7899352 5.271859e-03
Sqrt_ra_MA  -2.669027e+01  11.4379544 -2.3334825 1.962284e-02
Sqrt_sk_CA  -1.446164e+03 341.9219774 -4.2295133 2.341975e-05
Sqrt_rza_CA  2.437217e+03 784.5009368  3.1067106 1.891815e-03

Confusion Matrices:

Train:

p_data   0   1
     0 110  31
     1  24  77

Test:

p_data  0  1
     0 16 12
     1  2  2
```

**Cincinnati Bengals**

Coefficients:

```
                Estimate   Std. Error    z value    Pr(>|z|)
(Intercept)   6.210411e+02 1.990207e+03  0.3120485 0.755003632
pts_CA        1.309660e+03 4.473996e+02  2.9272724 0.003419492
sk_CA        -1.402161e+02 5.616404e+01 -2.4965450 0.012540977
ints_CA      -1.738445e+01 6.328826e+00 -2.7468683 0.006016729
pr_CA         4.327728e+03 1.477282e+03  2.9295195 0.003394865
td_CA        -8.727761e+03 3.343587e+03 -2.6102989 0.009046315
fgat_CA      -8.522874e+03 3.457652e+03 -2.4649310 0.013703962
pa_MA2       -1.676336e+00 6.289140e-01 -2.6654452 0.007688647
Sq_pts_CA    -1.334648e+01 4.683082e+00 -2.8499344 0.004372825
Sq_pc_CA      7.934078e-02 3.321110e-02  2.3889842 0.016895030
Sq_pr_CA     -3.024478e+02 9.873315e+01 -3.0632847 0.002189217
Sq_td_CA      7.857725e+02 3.098266e+02  2.5361687 0.011207272
Sq_fgat_CA    8.825961e+02 3.727883e+02  2.3675529 0.017906161
Sq_td_MA2    -1.268930e-01 3.945173e-02 -3.2164109 0.001298048
Sq_pen_CA2    7.983711e-04 2.972007e-04  2.6863026 0.007224763
Sqrt_pts_CA  -6.889970e+03 2.355644e+03 -2.9248779 0.003445914
Sqrt_sk_CA    4.003177e+02 1.695594e+02  2.3609298 0.018229180
Sqrt_pr_CA   -8.814628e+03 3.087625e+03 -2.8548252 0.004306052
Sqrt_td_CA    1.537159e+04 5.905615e+03  2.6028777 0.009244492
Sqrt_fgat_CA  1.422150e+04 5.708461e+03  2.4913022 0.012727583
Sqrt_pa_MA2   1.924016e+01 7.175164e+00  2.6814943 0.007329415
```

Confusion Matrices:

Train:

```
p_data  0  1
     0 98 35
     1 44 95
```

Test:

```
p_data  0  1
     0  2  0
     1 17 13
```