

LEGITIMATE OR UNFAIR?: AN EVALUATION AND IMPROVEMENT OF THE
COLLEGE FOOTBALL PLAYOFF USING LOGISTIC REGRESSION AND
ADJACENCY MATRICES

By

JERICO BOLOS LAWSON

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors Degree
With Honors in

Statistics & Data Science

THE UNIVERSITY OF ARIZONA

M A Y 2 0 2 0

Approved by:

Legitimate or Unfair?: An Evaluation and Improvement of the College Football Playoff Using Logistic Regression and Adjacency Matrices

Jericho Lawson
The University of Arizona
Faculty Advisor: Dr. Joseph Watkins
Thesis for B.S. in Statistics & Data Science with Honors

ABSTRACT. The world of college football has the unique challenge of picking the best teams in the football championship subdivision. Always up for debate, many fans, writers, and scholars have questioned whether the best teams in college football by season's end are truly the best teams. This research explores the history of finding the best college football teams since the beginning of the NCAA, including the current College Football Playoff. Because of the current method's subjectivity, a method consisting of four logistic regression models and a series of weights is used in an adjacency matrix to determine the best teams in the nation. The logistic regression models are based on game data from the top 25 teams during the first six seasons of play under the current College Football Playoff. With an average difference of 7.49 places between teams in both the College Football Playoff rankings and our rankings, our method only serves as a foundation for what could be a more objective way of determining which teams deserve to be at the top, particularly with logistic regression and more expansive game data.

Key Words: NCAA, College Football Playoff, Statistics, Bowl Championship Series, Logistic Regression, Adjacency Matrix.

1. Introduction

Picking the best teams in college football can be a huge challenge. In college sports, the College Football Playoff system has been criticized and debated constantly since its inception six years earlier, as well as its predecessors. Sports fans, writers, and even scholars have questioned whether the four best teams by season's end are truly the four best teams in college football. David H. Annis and Samuel S. Wu suggested that "the number of teams" influences the playoff procedure more so than the ranking procedure does [1]. This research will focus primarily on the College Football Playoff, as well as its predecessors, to examine the committee's methods for choosing the best teams to play in the championship and playoff. A revised version of the current playoff system with a new set of metrics and techniques will be used to reevaluate which teams from the last few seasons deserve to be in the playoff. The main question is the following:

How effective have the current and previous processes been for choosing the four best college football teams in the nation, and which metrics can be used to better select these teams using a more objective approach?

A brief history will be given regarding how the NCAA's idea of a playoff has changed since the inception of college football in 1859. From there, we will look at several research articles to investigate what work has been done regarding the topic of interest. Then, a discussion of data and

its collection will be explained, followed by some quick data analysis. After that, we will propose a new metric for selecting the best teams in the nation for the playoff.

2. History

Adam Augustyn from Encyclopaedia Britannica details the origins of the various methods to determine the best teams in the nation since the beginning of college football. Early methods of determining the best team were scattered and messy. In the past and in recent memory, the Associated Press has been the main authority for choosing the best team in college football. In the early days of NCAA football, there wasn't a playoff—the “best team” was chosen primarily based on the Associated Press and football coaches' polls [2]. The best team from a specific year based off these polls was awarded a championship. However, many other smaller organizations started designating titles, which led to some ambiguity and confusion regarding who the true champion was in college football each year [2]. The Helms Foundation, John Dickinson, Sporting News, and the National Football Foundation all had systems that attempted to pick and award a national champion [4]. Even as the population recognized the AP and Coaches' polls as true indicators of a championship, occasionally there were years when two different champions were crowned due to a disagreement as to who was the best team by both authorities.

The closest system for most of the 20th century that was similar to that of the current playoff was a series of bowl games. In an effort to “gain national recognition and economic growth through their football teams,” colleges and universities began organizing regional conferences and thereafter, crossover games with “regional prestige” at stake [10]. The first of these bowl games was the Rose Bowl, which annually began play in 1916 to determine “an unofficial national champion” [10]. Realizing the profitability of the Rose Bowl, other conferences and schools began to implement similar bowl games: the Orange Bowl in 1933, the Sugar Bowl in 1935, and the Cotton Bowl in 1937 [10]. Throughout the middle of the 20th century, more and more bowl games appeared. This only made the process of crowning a national champion more challenging. As a result, by the 1970s, the bowls' inability to determine a true national champion that everyone could agree upon was apparent. With many organizers resisting against a uniform championship playoff, a potential playoff system did not come together until 1992 with the creation of the Bowl Coalition.

The Atlantic Coast, Big East, Big 8, Southeastern, and Southwest conferences, as well as Notre Dame, all created the Bowl Coalition in 1992 in which a national champion could be crowned. [10]. The Cotton, Fiesta, Orange, and Sugar bowls now had a selection process for picking the best teams to play in those games. Additionally, the number one team from one of the five conferences would choose the bowl game they would play in, and then that team would play the second best team in the coalition.

However, three years later, the Bowl Alliance was instituted, which removed all bowl affiliations from their bowl games [2]. While an improvement was made, the Bowl Alliance, as well as its predecessor, were missing two of the biggest conferences nationally at the time: the Big Ten and Pacific-10 conferences. These two conferences still had their conference champions play in the Rose Bowl to determine the best college football team regionally. The lack of participation from these two conferences made the challenge of playing in a national championship less meaningful when the number-one team in the nation was potentially playing in the Rose Bowl.

After years of scrutiny, a true championship amongst all of the then-current Power-Five conferences (Big-Ten, Big-12, ACC, SEC, and Pac-10) was introduced with the creation of the Bowl Championship Series in 1998. Under this new agreement, the Rose, Orange, Sugar, and Fiesta

Bowls, as well as a new championship game, would showcase the best teams from all of the listed conferences and independent schools [3].

Participants in the championship game, which initially took place in one of the four aforementioned bowls until 2006, were determined by a computerized-ranking system that “gave equal weight to the AP poll, the coaches’ poll, and an average of six computer rankings” [3]. Now, instead of the polls directly determining the champion at season’s end, there was a dedicated game to determine the best team. However, this raised the question, “who should play in the national championship?” In its 16 years of existence, the BCS produced ten matchups of the two top-ranked teams in the nation. However, there were still discrepancies between what the polls decided the best teams were. This came to a head in 2003 when the University of Southern California was not selected to play in the national championship despite being ranked #1 in both the AP and Coaches’ polls. This was due to USC being ranked lower than expected in the computer rankings [3]. Consequentially, this led to the only split championship in the BCS and led to the Harris poll replacing the AP poll for determining who played in the national championship [3].

In 2014, the College Football Playoff was implemented to replace the BCS. In this system, there consists six bowl games (Rose, Sugar, Cotton, Orange, Fiesta, and Peach), two of which would be used for the four-team playoff [2]. A four-team bracket was implemented to determine a national champion. The winners of the two semi-final games would play in a national championship at a neutral site. While the previous system used computer rankings, the AP poll, the Coaches’ poll, and the Harris poll, the College Football Playoff used a 13-person committee composed of “high integrity football experts, with experience as coaches, student-athletes, college administrators and journalists, along with sitting athletics directors” [11]. To determine the best college football teams in the nation, the committee ranks the top 25 teams out of all possible teams in the NCAA Division-I Football Bowl Subdivision using “conference championships won, strength of schedule, head-to-head results, and comparison of results against common opponents” [11].

At the conclusion of the 2018 season, #1 seeds are 3-2 in semifinals and 0-3 in championships, #2 seeds are 4-1 in semifinals and 3-1 in championships, #3 seeds are 1-4 in semifinals and 0-1 in championships, and #4 seeds are 2-3 in semifinals and 2-0 in championships. As of September 2019, Rob Mullens, Gary Barta, Frank Beamer, Paola Boivin, Jo Catiglione, Ken Hatfield, Chris Howard, Ronnie Lott, Terry Mohajir, Ray Odierno, R.C. Slocum, Todd Stansbury, and Scott Stricklin make up the 13-person committee that ranks college football teams for the College Football Playoff. There is not a single statistician in the committee; all committee members are either current or former athletic directors, current or former head coaches, university presidents, journalism professors, former All-Americans, and chiefs of staff. Most of these individuals come from specific universities, creating implicit bias with ranking the best college football teams in the Football Bowl Subdivision.

Despite improvements to determining a clear-cut national champion, there is still a heavy amount of criticism as to whether the national champion is the rightful champion. Furthermore, there is some ridicule regarding which teams are selected to play in the playoff each year. For example, in 2017, the University of Alabama made the playoff despite not making it to the SEC championship, which roused many fans [6]. Nonetheless, the current football playoff system is still an improvement to methods done in the past, especially prior to 1992. Table 2.1 shows all of the official authorities for choosing a champion for a particular season in the Football Bowl Subdivision. Only recently has there been only one primary authority for choosing a champion. Much of the 20th century was filled with multiple authorities that picked champions, which resulted in up to four

Official Championship Authority	Years
National Championship Foundation (NCF)	1869-1935
Helms Athletic Foundation	1883-1935
College Football Researchers Association	1919-1935
Associated Press Poll	1936-1997, 2003
Coaches' Poll	1950-1995
Football Writers Association of America	1954-1997, 2003
National Football Foundation	1959-1997
USA/CNN/ESPN	1982-1997
Bowl Championship Series	1998-2013
College Football Playoff	2014-present

TABLE 2.1. List of Official Authorities for Awarding Championships in NCAA Football

champions in any given year [9]. As mentioned earlier, this only created ambiguity and confusion regarding who the best team was in NCAA football.

3. Literature Review

László Csató mentions the difficulties of ranking in generalized tournaments, specifically with regard to tennis. Ideally, a ranking system should include self-consistency and order preservation. Self-consistency involves assigning the same rank for players with equivalent results. Through this concept, a player that shows a better performance through wins in a tournament should be ranked strictly higher [5]. With order preservation, this prevents player A from being judged better in both the first and second halves of the season than player B; however, player A can be ranked lower on the basis of the whole season. Csató shows that there is incompatibility between self-consistency and order preservation. Furthermore, Csató mentions that no anonymous and neutral individual scoring method satisfies self-consistency [5]. In other words, if you re-index rounds, preserve the scores of the given players, and say that the scores are independent of the labeling of the players, this contradicts the idea of assigning the same rank to players with equivalent results. Additionally, Csató notes that no scoring method satisfies order preservation and self-consistency [5]. As a result, the author suggests that one has to wait until all tournament results are known in order to aggregate them and subsequently rank players.

Most previous research has attempted to show better methods for predicting and ranking the best teams in the nation, with a desired goal of bringing objectivity to the selection process in the College Football Playoff. Ilan Goodman, Kat Gregory, and Sunil Pai tried using a network-based approach to ranking college football teams in the championship subdivision. Unlike other computer-based ranking systems, theirs incorporated “garbage-time” detection to produce analogous weighted graphs to produce better rankings [7]. By training a model based on score, time remaining in a seasonal game, the down, distance, yard line, and similar features to win probability, they were able to define when a certain part of the game was considered garbage time. In their case, if the winning team had at least a 95% chance of winning the game from one point of the game towards the very end of the game, this was considered garbage time. To show its effectiveness, the method attempted to predict binary outcomes based on the current ranking prior to the week of play. From there, the machine learning results were compared to the true results. When included with Weighted Katz centrality, the predictive accuracy was 74.8% [7]. However, the authors found

that a BeatPower system, which is based solely off wins and losses, proved to be the better method of predicting which team was correctly ranked with a 86.3% prediction accuracy [7].

Kolbush and Sokol developed a logistic regression Markov chain model that was able to successfully rank teams in the Bowl Championship Series most of the time. Years prior, they did a similar model for the NCAA basketball tournament, which relied heavily on “home-and-home” match-ups to determine transition probabilities between ranked teams [8]. 15 seasons of college football from 2002-2016 were used to construct their model. Bowl games were used to determine the accuracy of the ranking system since they are mostly played on neutral fields, incorporates a full season worth of data to include in the model, and teams of similar strength play against each other in the bowl games. Overall, by examining the common opponents that teams play in a given season rather than look at historical team stats from prior seasons, they were able to find a new method for ranking college football teams, which was “among the best ranking systems in college football for predicting postseason games.”

4. Statistical Methods and their Usefulness

As Kolbush and Sokol did in their paper, a logistic regression model with Markov chains may be beneficial for finding a new metric for ranking college football teams during the season. A logistic regression model is useful for defining probabilities that a certain team will be ranked at a certain spot. On a more basic level, logistic regression can be useful when classifying binary or categorical responses. For instance, if we wanted to determine whether a team will win or lose based off certain factors, such as game score and time of possession, then we can use logistic regression to classify whether the game would be a win or a loss for that team. The following is the general equation for classifying a response for logistic regression:

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n, \quad (4.1)$$

where π is the response of the regression, β_i is the coefficient of the predictors, and x_i is the predictor.

The logistic regression method will be used moving forward to find predictors that can classify wins accurately and ultimately rank the best teams in college football.

5. Data Processing

Data is crucial in order to determine the success of my method. Various pieces of data will be used over the first six years of the College Football Playoff and combined into one giant dataset.

5.1. Data

A couple pieces of data will be used in determining how effective the College Football Playoff truly is and creating a new metric for ranking the best college teams in the nation. Firstly, the College Football Playoff final polls from 2014-2019 were gathered from the CFP’s official website, entered into Excel, and then imported into R. Additionally, all teams that were listed in the rankings for a particular season from 2014-2019 had specific game data, which included general game information (e.g. location, TV channel, day), offensive information for the team (e.g. passing yards, turnovers, penalties), and offensive information for the opponent (e.g. passing yards,

turnovers, penalties). Respectively, these pieces of data are found on the Official College Football Playoff website [12] and Sports Reference [13].

Having many variables of data associated to a game is beneficial in finding significant variables, and in turn, determining how teams are ranked once a season comes to an end. There are more variables to a game aside from the final score. In essence, the data that we are attempting to look for can be separated into three categories: general game information, team information, and opponent information. General game information can be defined as variables that relate to how the game is being set up. This involves information such as the game number, day, and team rank. Team information involves offensive game statistics related to the team at hand. This involves passing yards, rushing touchdowns, and fumbles. Likewise, opponent information involves offensive game statistics related to the opponent. Statistics include those mentioned for team information, but for the opponent. A detailed list of variables can be found in Appendix A. Aside from general game information, perhaps there are other offensive statistics associated to the finer workings of a single game that can help predict the top 25 teams generally. This is the motivation for including more than just general game information in our dataset.

As such, 59 different variables representing one of three different types were collected from 150 teams that were listed in the final College Football Playoff rankings from 2014 to 2019. Some general team information was found on the Official College Football Playoff website, while the rest of the data was found on Sports Reference. Through the use of R and the R packages `rvest`, `tidyr`, `dplyr`, we were able to stitch together all data points and make one huge dataset.

There are plenty of benefits and challenges with the use of this dataset, which we can call the “main” dataset. The main dataset has various information about how the two teams performed throughout the game; however, there is no information regarding how teams do progressively throughout the game. This means that our proposed model cannot take any in-game effects as the game goes on. For instance, there is no way to tell if one team outperformed another team during the third quarter of a game. In the end, only the game totals and averages matter since that is all we have in this dataset. Despite this, we can analyze how certain strategies were crucial to a team’s success or failure for each game. For example, we can see if more rushing attempts were a better indicator of a win versus more passing attempts. This alone makes the model more dependent on strategic factors rather than just the pure final score.

While we have game statistics for all top twenty-five schools in each of the six years of the College Football Playoff, we do not have game statistics compiled for the rest of the Division I Football Championship subdivision. This makes it rather difficult to make reliable decisions using logistic regression, which will be described later in the paper.

6. College Football Playoff Methodology

Many methods have been used to determine the effectiveness of ranking college football teams in the Division-I Football Championship Subdivision, such as using logistic regression and networks. Our method is loosely based off Kolbush and Sokol’s method of logistic regression Markov chain models, but is more focused on the probabilities found when using logistic regression.

Since we only have data from games in which at least one of the teams was in the top 25 in the College Football Playoff’s final polls for any season from 2014 to 2019, we felt it would be appropriate to split the games up into four different categories. Those categories are the following:

- Ranked team vs. ranked opponent (447 games)
- Ranked team vs. unranked opponent (937 games)

- Unranked team vs. ranked opponent (96 games)
- Unranked team vs. unranked opponent (381 games)

Not all games are created equal, and motivations are different for teams when they play against ranked teams versus unranked teams. Using the AP Poll ranks found in the Sports Reference data, we separated all games involving a top 25 team at the end of the year into those four categories. Thereafter, logistic regression will be used separately on each of the four different categories, creating four different models, each trying to predict the outcome for the team using most of the statistics in the dataset aside from team rank, opponent rank, and others. Using four models can potentially highlight different playing styles that occur when teams are unequal in strength; thus, changing the probabilities of winning or losing games. Additionally, on top of these four models, three more logistic regression models are included in which the result is predicted using only the team rank and opponent rank (when applicable). Why include these logistic regression results? Using the probabilities from the logistic regression model for most of the stats, as well as the probabilities from the logistic regression model using the ranks, a “score” is calculated for the two teams in an adjacency matrix. The adjacency matrix is $p \times p$, where p is the amount of unique teams found in the dataset, and contains scores and zeros for all elements in the matrix. When a prediction has been made on a certain game, a probability is computed for the team’s chance of winning in both the logistic regression model containing most statistics and the logistic regression model containing just the ranks. For each of the two models, the probabilities are altered depending on the true result of the game. If the specific team actually wins the game, the probability (of winning the game) calculated from the first model (i.e p_a) is subtracted from one and the probability calculated from the second model (i.e p_r) is also subtracted from one. Otherwise, the probability calculated from the first model is negated, while the probability of the second model is unchanged. Table 6.1 shows this derivation:

	Actual Win	Actual Loss
Score in Matrix	$(1 - p_a) \times (1 - p_r)$	$-p_a \times p_r$

TABLE 6.1. Derivation of Score in Adjacency Matrix

Because of this method, we use logistic regression based on only the rankings as well. In the case of games with only unranked teams, the unranked vs. unranked category will use a weight of 0.1 when computing a score to an element in the adjacency matrix.

We created the logistic regression models using the “glm” function in the programming language R. In those models, some of the original 59 variables were omitted due to redundancy with other variables in the data, potential cross-linearity, or lack of worth at second glance. The team name, opponent name, streak, date, time, amount of season wins, amount of season losses, average rushing yards for, total yards for, total average yards for, pass percentage for, number of plays for, turnovers for, total first downs for, team points, team rank, average rushing yards against, total yards against, total average yards against, pass percentage against, number of plays against, turnovers against, total first downs against, opponent points, and opponent rank were all omitted from the data for the logistic regression models containing most of the data. The team and opponent ranks were used for the second set of linear regression models.

After running initial linear regression models on games in the four different game categories, the “step” function was used to find the best set of predictors that would make the AIC as low as possible while still having high predictive success. Two categories had models that needed

to be reduced further, which will be talked about in the discussion section. Table 6.2 shows the predictors that are used in the logistic regression models moving forward:

	Ranked/Ranked	Ranked/Unranked	Unranked/Ranked	Unranked/Unranked
Variables	Pass.Att.For Pass.TD.For Rush.Att.For Rush.Yds.For Rush.TD.For Pass.FD.For Fumbles.For Int.For Pass.Att.Against Pass.TD.Against Rush.Att.Against Rush.Yds.Against Rush.TD.Against Pass.FD.Against Fumbles.Against Int.Against	Game Conference Pass.Att.For Pass.Yds.For Pass.TD.For Rush.Yds.For Rush.TD.For Penalty.FD.For Fumbles.For Int.For Pass.Cmp.Against Pass.Att.Against Pass.Yds.Against Pass.TD.Against Rush.Yds.Against Rush.TD.Against Penalties.Against Fumbles.Against Int.Against	Pass.TD.For Rush.TD.For Penalty.Yds.For Fumbles.For Pass.TD.Against Rush.TD.Against Penalty.FD.Against	Site Pass.TD.For Rush.TD.For Pass.FD.For Rush.FD.For Penalty.FD.For Fumbles.For Int.For Pass.TD.Against Rush.TD.Against Pass.FD.Against Rush.FD.Against Penalty.FD.Against Penalties.Against Fumbles.Against Int.Against

TABLE 6.2. Predictors for Main Logistic Regression Models in the Four Different Categories of Games

For each of the main four logistic regression models, we used the given variables mentioned in Table 6.2. While the “step” function is useful for verifying the best possible logistic regression model, we want to determine how accurate predictions would be based on the training data. For the 447 games identified as “ranked vs. ranked”, we get a prediction accuracy of 93.1%, a sensitivity (i.e. percentage of actual wins classified as wins) of 92.9%, and a specificity (i.e. percentage of actual losses classified as losses) of 93.3%. When predicting on the 937 games identified as “ranked vs. unranked” using its specific logistic regression model, there is a prediction accuracy of 97.3%, a sensitivity of 98.3%, and a specificity of 89.0%. For the 96 “unranked vs. ranked” games, we get a prediction accuracy of 88.5%, a sensitivity of 90.8%, and a specificity of 83.9%. Finally, for the 381 games that are classified as “unranked vs. unranked”, we retrieve a prediction accuracy of 97.1%, a sensitivity of 98.0%, and a specificity of 88.9%. When using a 0.5 threshold, prediction accuracies are relatively strong, hovering between 88 and 98 percent.

As mentioned earlier, all of the game data from the top 25 teams over the 2014-2019 seasons will be used as training data for the logistic regression models. The logistic regression models will be used to predict a game, based off whether the team and opponent are ranked, and a score will be assigned for the given team against the given opponent, as found in Table 6.1. These scores are placed into an adjacency matrix for a specific season. As a result, six different adjacency matrices carry all of the scores for each team that is found in the large dataset.

In order to rank the top 25 teams after each season, the sum of all columns in each row of the adjacency matrix can be seen as the total score for a specific team in each season. The total scores

are found and ordered as a way to rank the best college football teams in the Division-I Football Championship Subdivision.

7. Discussion & Results

7.1. The Logistic Regression Models

In total, seven different logistic regression models are used in our methodology for ranking the best college football teams in the nation. Aside from the “unranked vs. unranked” category, each category of games has two logistic regression models, which will be used as a probability and a weight: the main logistic regression model containing most statistics as predictors and the weight logistic regression model, which contains the teams’ AP rankings as predictors. Since the “unranked vs. unranked” games contain no rankings, a weight of 0.1 was set as the standard to replace what would be a weight logistic regression model. There are some noticeable patterns and observations that can be detected from the results of the logistic regression models.

7.1.1. “Ranked vs. Ranked” Models

After training on data from 447 games using the initial logistic regression model, the model was slimmed down to get to the lowest possible AIC. As a result, the following main logistic regression model in Equation 7.1 and Table 7.1 is used for predicting whether a team will win a certain game based off various statistics:

$$P = \frac{e^{0.210769 + \sum_{i=1}^{16} \beta_i x_i}}{1 + e^{0.210769 + \sum_{i=1}^{16} \beta_i x_i}} \quad (7.1)$$

Var.	Name	β Value	p-Val.	Var.	Name	β Value	p-Val.
x_1	Pass.Att.For	-0.127870	0.007	x_9	Pass.Att.Against	0.149381	0.001
x_2	Pass.TD.For	2.233300	$3 \cdot 10^{-8}$	x_{10}	Pass.TD.Against	-2.162224	$5 \cdot 10^{-8}$
x_3	Rush.Att.For	-0.094801	0.030	x_{11}	Rush.Att.Against	0.078803	0.055
x_4	Rush.Yds.For	0.024671	0.0001	x_{12}	Rush.Yds.Against	-0.023280	0.0001
x_5	Rush.TD.For	1.306737	$4 \cdot 10^{-5}$	x_{13}	Rush.TD.Against	-1.372697	$1 \cdot 10^{-5}$
x_6	Pass.FD.For	0.363019	0.003	x_{14}	Pass.FD.Against	-0.373491	0.002
x_7	Fumbles.For	-0.917675	0.004	x_{15}	Fumbles.Against	0.820860	0.015
x_8	Int.For	-1.186751	0.0004	x_{16}	Int.Against	1.018462	0.001

TABLE 7.1. Main Logistic Regression Model for “Ranked vs. Ranked” Games

No problems exist regarding the fit of the logistic regression model itself. All but one predictor is statistically significant, based off the α level of 0.05 and the null hypothesis $H_0 : \beta_i = 0$. As a result, we expect that these predictors can help find a reasonable probability for a team to win a game given the significant predictors seen in this model. Passing touchdowns, rushing touchdowns, and interceptions from both teams in a game have large coefficients that can influence the probability of a team winning a game. Considering that the result of the game is based off the amount of points a team can score, having large coefficients for these predictors makes plenty of sense. Touchdowns directly influence the amount of points a team has during the game.

Additionally, the team rank and opponent rank from the 447 games help determine the proper logistic regression model for classifying wins and losses, and consequently the weight assigned to

each and every game in this category. The following logistic regression model in Equation 7.2 is used to determine that weight:

$$W = \frac{e^{-0.15246 - 0.04428 \cdot TeamRank + 0.08443 \cdot OppRank}}{1 + e^{-0.15246 - 0.04428 \cdot TeamRank + 0.08443 \cdot OppRank}} \quad (7.2)$$

Both predictors have p-values below 0.05 and are therefore statistically significant. In this model, the opponent rank makes more of a difference in the weight of a game instead of team rank. Once again, there are no problems fitting this logistic regression model.

Overall, both models for this category have statistically significant coefficients that can allow for potentially successful winning probability and in turn, successful rankings.

7.1.2. “Ranked vs. Unranked” Models

Using similar methodology, the best main logistic regression model is found for the 937 games that were classified as “ranked vs. unranked”. Equation 7.3 and Table 7.2 showcase the optimal logistic regression model, which contains 19 different predictors:

$$P = \frac{e^{0.08456 + \sum_{i=1}^{19} \beta_i x_i}}{1 + e^{0.08456 + \sum_{i=1}^{19} \beta_i x_i}} \quad (7.3)$$

Var.	Name	β Value	p-Val.	Var.	Name	β Value	p-Val.
x_1	Game	0.2169	0.039	x_6	Rush.Yds.For	0.02445	$6 \cdot 10^{-5}$
x_{2a}	ConferenceAmerican	1.43	0.215	x_7	Rush.TD.For	2.576	$4 \cdot 10^{-7}$
x_{2b}	ConferenceBig 12	1.817	0.072	x_8	Penalty.FD.For	-0.6374	0.004
x_{2c}	ConferenceBig Ten	-0.7889	0.412	x_9	Fumbles.For	-1.253	0.0004
x_{2d}	ConferenceCUSA	13.7	0.995	x_{10}	Int.For	-1.661	$5 \cdot 10^{-5}$
x_{2e}	ConferenceInd	2.572	0.121	x_{11}	Pass.Cmp.Against	-0.3562	0.003
x_{2f}	ConferenceMAC	2.291	0.780	x_{12}	Pass.Att.Against	0.3343	0.0001
x_{2g}	ConferenceMWC	2.358	0.091	x_{13}	Pass.Yds.Against	-0.02601	$7 \cdot 10^{-5}$
x_{2h}	ConferenceNon-Major	13.23	0.992	x_{14}	Pass.TD.Against	-3.464	$6 \cdot 10^{-8}$
x_{2i}	ConferencePac-12	-0.6686	0.459	x_{15}	Rush.Yds.Against	-0.03489	$3 \cdot 10^{-6}$
x_{2j}	ConferenceSEC	-2.695	0.008	x_{16}	Rush.TD.Against	-2.756	$5 \cdot 10^{-7}$
x_{2k}	ConferenceSun Belt	0.7557	0.624	x_{17}	Penalties.Against	0.5543	0.0004
x_3	Pass.Att.For	-0.1311	0.008	x_{18}	Fumbles.Against	2.016	$7 \cdot 10^{-6}$
x_4	Pass.Yds.For	0.0305	$4 \cdot 10^{-5}$	x_{19}	Int.Against	1.614	0.0004
x_5	Pass.TD.For	2.357	$1 \cdot 10^{-6}$				

TABLE 7.2. Main Logistic Regression Model for “Ranked vs. Unranked” Games

One categorical predictor remains in this logistic regression model: the opponent’s conference. Within the conference predictor, only one conference stands out as statistically significant: the SEC. From fans of college football, this only cements the mindset of many: the fact that the SEC is seen as the superior conference in Division-I college football. From a statistical point of view, an opponent from the SEC conference lowers the chance of a particular team to win significantly, with a p-value of 0.008. Aside from this categorical predictor, all 18 of the other predictors are statistically significant. Passing yards, passing touchdowns, rushing yards, and rushing touchdowns

from both teams, as well as interceptions thrown by the team and fumbles lost by the opponent, had a dramatic effect on the probability of a team winning against the opponent. No problems exist when trying to fit this model, suggesting that this model is legitimate for classifying wins and losses for specific games in the “ranked vs. unranked” category. As mentioned earlier, there is a 97.3% prediction accuracy when classifying wins and losses on the training data (i.e. the 937 games in this category).

Additionally, only the team rank from the 937 games determine the proper logistic regression model for classifying wins and losses, and consequently the weight assigned to each and every game in this category. Since there is no opponent rank for this category, we must rely only on the team rank. As a result, the following logistic regression model in Equation 7.4 is used to determine that weight:

$$W = \frac{e^{2.41976 - 0.02756 \cdot TeamRank}}{1 + e^{2.41976 - 0.02756 \cdot TeamRank}} \quad (7.4)$$

The team rank is technically not a significant predictor with an α level of 0.05. However, the p-value of this predictor is 0.0596, suggesting that the coefficient does lead to some possible change. When looking at the coefficient estimate of the team rank predictor, the negative sign makes sense intuitively. If a team is ranked lower, it makes sense that the team will have a more difficult time trying to win against the opponent. Aside from the relatively high p-value for the predictor, there are no other problems that exist with this model fit.

While the main logistic regression model provides evidence of strong significant predictors, the weight logistic regression model leads more to desire. This in turn, may lead to some problems when predicting the rankings of the top 25 college football teams in the nation.

7.1.3. “Unranked vs. Ranked” Models

As with the last two categories, the best main logistic regression model is found for the 96 games classified as “unranked vs. ranked”. However, some considerable issues appear when finding this model initially. The original logistic regression model talked about in the methodology section of this paper already suffers from two warning messages for this category. The algorithm did not converge and either probabilities of zero or one occurred. This can cause problems for smaller logistic regression models based off the original model, even with the use of the “step” function in R. To resolve this issue, we try to reduce the amount of predictors by looking at the correlation plots and removing any predictors that have high correlations with one another. As a result, we reduce the model from 12 predictors to seven and achieve a successful model that is able to give significant results. Equation 7.5 and Table 7.3 show the model that has the lowest AIC and a reduction in predictors:

$$P = \frac{e^{1.58145 + \sum_{i=1}^7 \beta_i x_i}}{1 + e^{1.58145 + \sum_{i=1}^7 \beta_i x_i}} \quad (7.5)$$

No problems exist with the fit of this model. All predictors except for fumbles are significant, based off a p-value that is less than 0.05. Like the other models, passing touchdowns and rushing touchdowns are important predictors for determining a win in this category. Additionally, penalty yards and fumbles are also significant variables for finding wins. As such, we will use this model to assign probabilities of winning games in order to assign a score to each game.

Var.	Name	β Value	p-Val.	Var.	Name	β Value	p-Val.
x_1	Pass.TD.For	2.40672	$6 \cdot 10^{-5}$	x_5	Pass.TD.Against	-1.89315	0.0002
x_2	Rush.TD.For	1.55012	0.0006	x_6	Rush.TD.Against	-2.18205	0.0006
x_3	Penalty.Yds.For	-0.04194	0.0401	x_7	Penalty.FD.Against	1.06183	0.0007
x_4	Fumbles.For	-0.64207	0.2179				

TABLE 7.3. Main Logistic Regression Model for “Unranked vs. Ranked” Games

With the main model providing helpful insights into classifying games as wins or losses, the weight model also leads to a useful result, as seen in Equation 7.6:

$$W = \frac{e^{-0.71133+0.11070 \cdot OppRank}}{1 + e^{-0.71133+0.11070 \cdot OppRank}} \quad (7.6)$$

Since no team rank applies in this category, the opponent rank is the only predictor used in this weight model. The opponent rank has a coefficient that is statistically significant with a p-value of 0.000893. Additionally, the model indicates that a higher opponent rank will lead to a surprisingly higher chance of winning games. Intuitively, this makes no sense. However, due to the nature of this data, the games in this category came from future top teams. In order to get to the top, these unranked teams would most likely have to win against higher ranked teams. Without any data from teams that did not eventually get to the top 25, this model can be insufficient to providing a reasonable weight for the probability that the main logistic regression model produces. As such, results from the weight model make the prospect of classifying wins and creating rankings challenging, but possible. With a successful main logistic regression model, we can still assign scores to games in this category despite the lack of data for other games in this segment.

7.1.4. “Unranked vs. Unranked” Models

Using the 381 games in the “unranked vs. unranked” category, a main logistic regression model is found. However, faults similar to the ones found in the model from the “unranked vs. ranked” category are found here. While the algorithm does converge, probabilities of winning are either zero or one. While there are more games in this category, the algorithm finds it hard to make probabilities for winning games. As a result, we reduce the amount of predictors in this model using correlation plots. Consequently, we reduce the amount of predictors from 24 to 16, which solves the case of high p-values and standard errors. As such, the best model is found, as seen in Equation 7.7 and Table 7.4:

$$P = \frac{e^{-6.02021+\sum_{i=1}^{16} \beta_i x_i}}{1 + e^{-6.02021+\sum_{i=1}^{16} \beta_i x_i}} \quad (7.7)$$

In the reduced model, all but four predictors are significant predictors. There are more predictors that are significant, including passing, rushing, and penalty first downs. While passing and rushing touchdowns have been seen in other models, it is worthwhile to note that first downs are significant in this model, suggesting that first downs are more important in games involving two unranked teams. Nonetheless, no problems exist with the fit of the model, which paves the way for the logistic regression model to classify wins and give probabilities that can be used for scores in our adjacency matrices.

Var.	Name	β Value	p-Val.	Var.	Name	β Value	p-Val.
x_{1a}	Site@	-1.72810	0.0225	x_9	Pass.TD.Against	-2.49774	$2 \cdot 10^{-5}$
x_{1b}	SiteN	-1.90398	0.1866	x_{10}	Rush.TD.Against	-2.11380	$4 \cdot 10^{-5}$
x_2	Pass.TD.For	1.37728	0.0004	x_{11}	Pass.FD.Against	0.09051	0.4132
x_3	Rush.TD.For	1.00754	0.0125	x_{12}	Rush.FD.Against	0.21583	0.0349
x_4	Pass.FD.For	0.22586	0.0150	x_{13}	Penalty.FD.Against	0.52416	0.0632
x_5	Rush.FD.For	0.50739	$9 \cdot 10^{-5}$	x_{14}	Penalties.Against	0.39287	0.0269
x_6	Penalty.FD.For	-0.31812	0.3287	x_{15}	Fumbles.Against	0.55488	0.2474
x_7	Fumbles.For	-0.98972	0.0173	x_{16}	Int.Against	1.88188	0.0002
x_8	Int.For	-1.02373	0.0086				

TABLE 7.4. Main Logistic Regression Model for “Unranked vs. Unranked” Games

While the other three categories have weight logistic regression models, we have to instead assign a generic weight for probabilities in this category in order to come up with this score. That is because there is no rank data for unranked teams and opponents, as defined by this category. The generic weight used for this category is set to 0.1. While a generic weight is used for this category, more research would need to be done to find the ideal generic weight for this category. Nonetheless, the main logistic regression model seen here proves to be successful.

7.2. Analysis of Results

Appendices B-G contain results of how our method predicted the rankings of the top 25 teams in each of the six final College Football Playoff rankings, as determined by the College Football Playoff Committee. By first glance, most of the teams in the top half of the initial rankings were projected to be ranked lower in our ranking system, while the opposite happened with teams in the bottom half of the initial rankings.

On average, the predicted rankings were off by 7.49 places from the true rankings set forth by the College Football Playoff Committee. As a result, it is hard to say that our method had great success when ranking teams from the first six seasons in the new format for the NCAA Division-I Football Championship Subdivision. 8 of the 150 teams, or 5.3%, in those polls were ranked perfectly between the CFP ranking system and our ranking system. However, 50 of the 150 teams, or 33.3%, in those polls had differences of ten or greater between rankings in the CFP ranking system and our ranking system.

There are many shortcomings that attribute to the unsuccessful results of our methods. Only game statistics of the top 25 teams were gathered from the first six seasons instead of all teams in Division-I college football. As a result, we do not get a full picture of how various statistics may influence the result of a particular game, and consequently, the ranking of a particular team. To compensate for that, our method attempted to create four separate logistic regression models representing the type of teams playing each other. While the models seemed promising, that did not translate into determining scores in the adjacency matrices. Additionally, with our methodology, teams that start the year off as one of the top ten teams in college football and that only play ranked teams throughout the year may find it hard to be at the top of the rankings by the end of the year. If the top teams in the league play against teams worse than them and win, then those teams do not benefit as much with regard to their score due to the high probability of them winning. However, if those top teams lose even one game to one of those teams, their score will dramatically decrease because a stronger weight is placed on a loss to a weaker opponent. In essence, our method puts

the top teams at an already difficult spot. They must maintain near-perfect records to stay at the top. However, if a team starts off the year unranked, achieving wins against the top teams in the nation will propel them up in the rankings. As a result, our method rewards teams that improve dramatically throughout the season, while penalizing teams that already start at the top of the AP rankings.

7.3. Challenges and Future Directions

With the huge discrepancies in the rankings between the College Football Playoff polling and our methodology, there are some challenges that occurred during the entire investigation. While the investigation focuses on only the top 25 college football teams in the Division-I Football Championship Subdivision, it leaves a grand majority of the other teams in that subdivision out of the equation. What if there was a non-top 25 team that had better in-game statistics that would propel them into the top 25? With the lack of game data for non-top 25 teams in this investigation, a future direction would be to gather game data for all teams in the D-I Football Championship Subdivision, which would eliminate the need to make four separate categories for the dataset and consequently have four different logistic regression models. Additionally, the potential logistic regression model would be more representative of the whole subdivision rather than the top teams within it.

Another challenge involved the use of weights and how to implement them into the creation of a score for the adjacency matrix. While having weights is important for distinguishing important games from the others, the probabilities of our potential logistic regression model could serve as the score that can be added into the adjacency matrix. Using AP rankings, which go against the grain of this paper, may not be the best method for finding weights. Additionally, some more research can be done in order to find weights that may be appropriate to include into the scores of each of the games. With regard to the score, a future direction would be to find other equations that may better determine a score in the matrix aside from the equations seen in Table 6.1.

If the methodology is kept the same, more data would be useful to finding better correlations and significant predictors. The “unranked vs. ranked” category only had 96 games within the category. Because of the relatively short tenure of the current college football playoff system, it was challenging to find more games in those categories unless we took data from years in which the BCS was implemented or wait until another time for more college football data to come in. Nonetheless, more data will be crucial to the success of this method.

Keeping with the theme of data collection, having more game statistics may be useful for finding other variables that may influence wins in college football. This may include largest score margin, weather conditions, and days of rest in between games. There are many more factors that exist aside from the offensive statistics, and as a future direction, they may be useful in improving the logistic regression model and ultimately the ranking system found in this paper.

While logistic regression is a popular machine learning classification technique, there are other methods that can be used to classify wins and losses (and ultimately probabilities of winning) that may produce better prediction accuracies and rankings. A future direction would be to try out other classification methods, such as linear discriminant analysis, quadratic discriminant analysis, and boosting.

Another future direction would be to use evaluation techniques to identify the usefulness of various predictors in the classification models. For instance, a quick check of the correlations among all the predictors would be useful for all models. Additionally, more evaluation techniques on

the classification models aside from the prediction accuracy would be useful. The “step” function allowed us to find the best model with the lowest AIC; however, diagnosing the residuals and normality of the classification fits would lead to a more informed choice of a proper model.

Finally, a future direction would be to investigate more trends with the dataset through exploratory data analysis. Due to the constraints of this project, not much exploratory data analysis was done on the data. The analysis can allow us to see more trends and find general characteristics in the dataset.

8. Conclusion

In summary, this paper provides a potential method for trying to rank the best teams in the Division-I Football Championship Subdivision by using logistic regression on four different categories of games and creating weights that would be used in an adjacency matrix. Since the logistic regression models had success in finding worthwhile predictors, this method, with the right amount of fixes, can become a potential predictor of future rankings down the line. More data, better machine learning techniques, and a streamlined model can improve upon the methods seen here. From the logistic regression models, passing and rushing touchdowns from both teams were important predictors in determining wins, as well as interceptions and fumbles.

Aside from the methodology, we have also explored the history of declaring the best team in college football, as well as some of the flaws that have been experienced along the way in the NCAA. The process of picking the best teams in college football does not have to be subjective, yet the NCAA’s use of a committee to make these decisions hinder the potential for an algorithm to properly rank the best teams without any bias. At the end of the day, using machine learning methods like the one seen in this paper can potentially provide a successful model that can rank the rightful teams in the College Football Playoff.

9. Acknowledgements

This project is one of the requirements for completing a Bachelor of Science in Statistics & Data Science with Honors through the Honors College and the Department of Mathematics at The University of Arizona. Special thanks is given to my faculty advisor Dr. Joseph Watkins, who helped establish guidelines and mentored me and my work throughout the entire project process.

10. Additional Notes

The dataset used for this project can be found through this URL:
<https://www.kaggle.com/jericholawson/ncaa-football-game-stats-of-top-25-teams-201419>

References

- [1] David H Annis and Samuel S Wu. A comparison of potential playoff systems for NCAA I-A football. *The American Statistician*, 60(2):151–157, 2006.
- [2] Adam Augustyn. The arrival of the college football playoff. 2015.
- [3] Adam Augustyn. Bcs. 2019.
- [4] Jesse Collins. The quest for a champion part II: The national championship become less mythical. 2013.

- [5] Laszlo Csato. Some impossibilities of ranking in generalized tournaments. *International Game Theory Review*, 21, 03 2019.
- [6] Chuck Culpepper. Alabama sneaks into college football playoff despite not making conference championship game. 2017.
- [7] Ilan N. Goodman, Kat Gregory, and Sunil Pai. A network-based approach to ranking college football teams. 2015.
- [8] J. Kolbush and J. Sokol. A logistic regression/markov chain model for American college football. *International Journal of Computer Science in Sport*, 16(3):185–196, 2017.
- [9] ncaa.com. College football championship history. 2019.
- [10] Michael Oriard. Gridiron football. 2019.
- [11] College Football Playoff. Overview. 2019.
- [12] College Football Playoff. Rankings. 2020.
- [13] Sports Reference. College football statistics and history. 2020.

Appendix A. List of Variables

Type	Variables
General Game Information	Game, Time, Day, Team, Site, Opponent, Conference, Result, Team Points, Opponent Points, Season Wins, Season Losses, Streak, TV Network, Team Rank, Opponent Rank, and Date
Team Information	Passing Completions, Passing Attempts, Passing Percentage, Passing Yards, Passing Touchdowns, Rushing Attempts, Rushing Yards, Average Rushing Yards, Rushing Touchdowns, Plays, Total Yards, Total Average Yards, Passing First Downs, Rushing First Downs, Penalty First Downs, Total First Downs, Penalties, Penalty Yards, Fumbles, Interceptions, and Turnovers
Opponent Information	Passing Completions, Passing Attempts, Passing Percentage, Passing Yards, Passing Touchdowns, Rushing Attempts, Rushing Yards, Average Rushing Yards, Rushing Touchdowns, Plays, Total Yards, Total Average Yards, Passing First Downs, Rushing First Downs, Penalty First Downs, Total First Downs, Penalties, Penalty Yards, Fumbles, Interceptions, and Turnovers

TABLE A.1. List of Variables in Giant Dataset Used for Logistic Regression Method

Appendix B. 2014 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
Alabama	1	9	-8	0.041721116
Oregon	2	12	-10	-0.039816162
Florida State	3	1	+2	0.447550183
Ohio State	4	19	-15	-0.275171183
Baylor	5	17	-12	-0.162954736
Texas Christian	6	6	0	0.129919366
Mississippi State	7	8	-1	0.075361539
Michigan State	8	13	-5	-0.060848338
Mississippi	9	20	-11	-0.306453019
Arizona	10	4	+6	0.179685279
Kansas State	11	2	+9	0.282534421
Georgia Tech	12	22	-10	-0.390444623
Georgia	13	23	-10	-0.521469359
UCLA	14	24	-10	-0.720632541
Arizona State	15	11	+4	-0.010995554
Missouri	16	7	+9	0.082436488
Clemson	17	15	+2	-0.118099097
Wisconsin	18	16	+2	-0.127404846
Auburn	19	21	-2	-0.316974334
Boise State	20	10	+10	0.024254256
Louisville	21	25	-4	-0.85945486
Utah	22	5	+17	0.172717331
LSU	23	3	+20	0.281034443
USC	24	18	+6	-0.271059565
Minnesota	25	14	+11	-0.077352427

TABLE B.1. Comparison of College Football Rankings and Our Method's Rankings for the 2014 Season Among the Top 25 Teams

Appendix C. 2015 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
Clemson	1	10	-9	0.022349203
Alabama	2	11	-9	0.007565858
Michigan State	3	2	+1	0.088992254
Oklahoma	4	5	-1	0.06105303
Iowa	5	16	-11	-0.035883252
Stanford	6	8	-2	0.048151537
Ohio State	7	21	-14	-0.329867028
Notre Dame	8	3	+5	0.087473076
Florida State	9	23	-14	-0.367254007
North Carolina	10	14	-4	-0.029293477
Texas Christian	11	6	+5	0.059785278
Mississippi	12	20	-8	-0.140268832
Northwestern	13	4	+9	0.06336798
Michigan	14	14	0	-0.011092718
Oregon	15	9	+6	0.027499079
Oklahoma State	16	19	-3	-0.122907359
Baylor	17	25	-8	-0.784038533
Houston	18	7	+11	0.058348817
Florida	19	1	+18	0.090010225
LSU	20	12	+8	-0.00348698
Navy	21	13	+8	-0.003511338
Utah	22	18	+4	-0.114413583
Tennessee	23	24	-1	-0.47905021
Temple	24	17	+7	-0.05408529
USC	25	22	+3	-0.347913775

TABLE C.1. Comparison of College Football Rankings and Our Method's Rankings for the 2015 Season Among the Top 25 Teams

Appendix D. 2016 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
Alabama	1	5	-4	0.14872867
Clemson	2	1	+1	0.305328636
Ohio State	3	21	-18	-0.644913925
Washington	4	9	-5	0.011374365
Penn State	5	6	-1	0.131130331
Michigan	6	16	-10	-0.11909361
Oklahoma	7	20	-13	-0.387926183
Wisconsin	8	4	+4	0.161311338
USC	9	13	-4	-0.023926691
Colorado	10	17	-7	-0.120558657
Florida State	11	24	-13	-0.82479292
Oklahoma State	12	8	+4	0.012286883
Louisville	13	12	+1	-0.004959317
Auburn	14	22	-8	-0.748506383
Western Michigan	15	10	+5	0.010248706
West Virginia	16	11	+5	-0.000105606
Florida	17	2	+15	0.218250895
Stanford	18	7	+11	0.058935496
Utah	19	19	0	-0.334764017
LSU	20	25	-5	-1.028101061
Tennessee	21	3	+18	0.215499303
Virginia Tech	22	15	+7	-0.081061452
Pittsburgh	23	23	0	-0.801068059
Temple	24	14	+10	-0.051835575
Navy	25	18	+7	-0.29925183

TABLE D.1. Comparison of College Football Rankings and Our Method's Rankings for the 2016 Season Among the Top 25 Teams

Appendix E. 2017 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
Clemson	1	21	-20	-0.46895431
Oklahoma	2	18	-16	-0.373031325
Georgia	3	7	-4	0.002763825
Alabama	4	10	-6	-0.038160372
Ohio State	5	6	-1	0.005909106
Wisconsin	6	5	+1	0.006514852
Auburn	7	24	-17	-0.574018586
USC	8	15	-7	-0.188338297
Penn State	9	22	-13	-0.499543789
Miami (FL)	10	12	-2	-0.068549941
Washington	11	19	-8	-0.416783812
UCF	12	3	+9	0.185790063
Stanford	13	13	0	-0.101432826
Notre Dame	14	8	+6	-0.004835854
Texas Christian	15	9	+6	-0.013954215
Michigan State	16	1	+15	0.257076169
LSU	17	25	-8	-0.640309846
Washington State	18	2	+16	0.228027272
Oklahoma State	19	11	+8	-0.038542823
Memphis	20	16	+4	-0.216228057
Northwestern	21	17	+4	-0.342255727
Virginia Tech	22	4	+18	0.080558695
Mississippi State	23	23	0	-0.544263753
North Carolina State	24	14	+10	-0.13490451
Boise State	25	20	+5	-0.444340392

TABLE E.1. Comparison of College Football Rankings and Our Method's Rankings for the 2017 Season Among the Top 25 Teams

Appendix F. 2018 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
Alabama	1	1	0	0.337626902
Clemson	2	10	-8	0.004216047
Notre Dame	3	9	-6	0.010119835
Oklahoma	4	18	-14	-0.251500725
Georgia	5	19	-14	-0.378635597
Ohio State	6	3	+3	0.204825177
Michigan	7	12	-5	-0.002821066
UCF	8	11	-3	0.000066646
Washington	9	24	-15	-0.892114897
Florida	10	8	+2	0.016861583
LSU	11	2	+9	0.324469176
Penn State	12	25	-13	-0.948881768
Washington State	13	7	+6	0.033791905
Kentucky	14	6	+8	0.079180656
Texas	15	5	+10	0.337626902
West Virginia	16	21	-5	-0.640500625
Utah	17	15	+2	-0.056471564
Mississippi State	18	13	+5	-0.033432261
Texas A&M	19	14	+5	-0.045863241
Syracuse	20	16	+4	-0.070743332
Fresno State	21	4	+17	0.189714128
Northwestern	22	20	+2	-0.55769064
Missouri	23	23	0	-0.777581147
Iowa State	24	17	+7	-0.210713304
Boise State	25	22	+3	-0.640503834

TABLE F.1. Comparison of College Football Rankings and Our Method's Rankings for the 2018 Season Among the Top 25 Teams

Appendix G. 2019 Predictions

Team	CFP Rank	Predicted Rank	+/- Difference	Score
LSU	1	2	-1	0.213899478
Ohio State	2	13	-11	0.005310173
Clemson	3	12	-9	0.007597218
Oklahoma	4	5	-1	0.06463674
Georgia	5	7	-2	0.054022667
Oregon	6	24	-18	-0.762737995
Baylor	7	20	-13	-0.102907307
Wisconsin	8	17	-9	-0.052186858
Florida	9	21	-12	-0.22858719
Penn State	10	14	-4	0.002160842
Utah	11	18	-7	-0.077306787
Auburn	12	1	+11	0.58833898
Alabama	13	23	-10	-0.726751012
Michigan	14	15	-1	-0.00363241
Notre Dame	15	4	+11	0.085084159
Iowa	16	6	+10	0.059926391
Memphis	17	22	-5	-0.450810972
Minnesota	18	16	+2	-0.012271174
Boise State	19	11	+8	0.01224051
Appalachian State	20	8	+12	0.02473167
Cincinnati	21	19	+2	-0.087130347
USC	22	25	-3	-0.950051826
Navy	23	10	+13	0.016094534
Virginia	24	3	+21	0.085771203
Oklahoma State	25	9	+16	0.023304899

TABLE G.1. Comparison of College Football Rankings and Our Method's Rankings for the 2019 Season Among the Top 25 Teams

Appendix H. Code

```

# Jericho Lawson
# Spring 2020
# Thesis Code

## This code uses logistic regression models and an adjacency matrix in order
## to rank the best 25 college football teams in the nation from 2014-2019. The
## "all_game_stats.csv" file contains all games played by the top 25 teams in
## the 2014-19 seasons.

## CONSTANTS & LIBRARIES

# Used for the unranked vs. unranked games as a weight.
UNRANKED_WEIGHT = 0.1

# Library for correlation plot.
library(corrplot)

## SETUP

# Set the working directory and import the .csv file.
setwd("C:/Users/mario/Documents/Honors Thesis/Code/")
all_game_stats = read.csv("all_game_stats.csv", header = T)

# Split into four categories: ranked vs ranked, unranked vs ranked,
# ranked vs unranked, and unranked vs unranked
rvr_ags = all_game_stats[which(is.na(all_game_stats$TeamRank) != T &
                             is.na(all_game_stats$OppRank) != T), -15] # 447
uvr_ags = all_game_stats[which(is.na(all_game_stats$TeamRank) == T &
                             is.na(all_game_stats$OppRank) != T), -15] # 96
rvu_ags = all_game_stats[which(is.na(all_game_stats$TeamRank) != T &
                             is.na(all_game_stats$OppRank) == T), -15] # 937
uvu_ags = all_game_stats[which(is.na(all_game_stats$TeamRank) == T &
                             is.na(all_game_stats$OppRank) == T), -15] # 381

## LOGISTIC REGRESSION MODELS (for probabilities)

# Finds initial logistic regression model for ranked vs. ranked games.
lm_1rvr = glm(Result ~ . - X - Team - Opponent - Streak - Date - Time - W - L -
              Rush.Avg.Yds.For - Rush.Avg.Yds.Against - Total.Yds.For -
              Total.Avg.Yds.For - Pass.Pct.For - Pass.Pct.Against - Plays.For -
              Plays.Against - Total.Yds.Against - Total.Avg.Yds.Against -
              Turnovers.For - Turnovers.Against - Total.FD.For -
              Total.FD.Against - TeamPts - OppPts - TeamRank - OppRank,
              data = rvr_ags, family = "binomial")

# Finds initial logistic regression model for unranked vs. ranked games.
# Removes TeamRank column due to NAs.
uvr_ags = uvr_ags[,-15]
lm_1uvr = glm(Result ~ . - X - Team - Opponent - Streak - Date - Time - W - L -
              Rush.Avg.Yds.For - Rush.Avg.Yds.Against - Total.Yds.For -
              Total.Avg.Yds.For - Pass.Pct.For - Pass.Pct.Against - Plays.For -
              Plays.Against - Total.Yds.Against - Total.Avg.Yds.Against -
              Turnovers.For - Turnovers.Against - Total.FD.For -
              Total.FD.Against - TeamPts - OppPts - OppRank,
              data = uvr_ags, family = "binomial")

# Finds initial logistic regression model for ranked vs. unranked games.
# Removes OppRank column due to NAs.
rvu_ags = rvu_ags[,-16]
lm_1rvu = glm(Result ~ . - X - Team - Opponent - Streak - Date - Time - W - L -
              Rush.Avg.Yds.For - Rush.Avg.Yds.Against - Total.Yds.For -
              Total.Avg.Yds.For - Pass.Pct.For - Pass.Pct.Against - Plays.For -

```

```

    Plays.Against - Total.Yds.Against - Total.Abg.Yds.Against -
    Turnovers.For - Turnovers.Against - Total.FD.For -
    Total.FD.Against - TeamPts - OppPts - TeamRank,
    data = rvu_ags, family = "binomial")

# Finds initial logistic regression model for unranked vs. unranked games.
# Removes TeamRank and OppRank columns due to NAs.
uvu_ags = uvu_ags[,-c(15, 16)]
lm_1uvu = glm(Result ~ . - X - Team - Opponent - Streak - Date - Time - W - L -
    Rush.Avg.Yds.For - Rush.Avg.Yds.Against - Total.Yds.For -
    Total.Abg.Yds.For - Pass.Pct.For - Pass.Pct.Against - Plays.For -
    Plays.Against - Total.Yds.Against - Total.Abg.Yds.Against -
    Turnovers.For - Turnovers.Against - Total.FD.For -
    Total.FD.Against - TeamPts - OppPts,
    data = uvu_ags, family = "binomial")

# Uses step function to find best possible logistic regression model for
# each category.
step(lm_1rvr, direction = "both")
step(lm_1uvr, direction = "both")
step(lm_1rvu, direction = "both")
step(lm_1uvu, direction = "both")

# Creates revised logistic regression model and assesses the accuracy of the
# model. Ranked vs. ranked.
lm_1rvrr = glm(formula = Result ~ Pass.Att.For + Pass.TD.For + Rush.Att.For +
    Rush.Yds.For + Rush.TD.For + Pass.FD.For + Fumbles.For +
    Int.For + Pass.Att.Against + Pass.TD.Against +
    Rush.Att.Against + Rush.Yds.Against + Rush.TD.Against +
    Pass.FD.Against + Fumbles.Against + Int.Against,
    family = "binomial", data = rvr_ags)
resultsvrr = predict(lm_1rvrr, rvr_ags[, c(19, 22, 23, 24, 26, 30,
    36, 37, 40, 43, 44, 45, 47,
    51, 57, 58)], type = "response")
resultsvrr = ifelse(resultsvrr > 0.5, 'W', 'L')
table(rvr_ags$Result, resultsvrr)

# Creates revised logistic regression model and assesses the accuracy of the
# model. Unranked vs. ranked.
lm_1uvrr = glm(formula = Result ~ Pass.Yds.For + Pass.TD.For + Rush.Yds.For +
    Rush.TD.For + Penalty.Yds.For + Fumbles.For + Pass.Att.Against +
    Pass.Yds.Against + Pass.TD.Against + Rush.Yds.Against +
    Rush.TD.Against + Penalty.FD.Against,
    family = "binomial", data = uvr_ags)

# Looks at correlations.
corrplot(cor(uvr_ags[,c(21, 25, 34, 35, 42, 46, 52)]))

# Redo model with predictors with high coefficients (above 100).
lm_1uvrr = glm(formula = Result ~ Pass.TD.For +
    Rush.TD.For + Penalty.Yds.For + Fumbles.For + Pass.TD.Against +
    Rush.TD.Against + Penalty.FD.Against,
    family = "binomial", data = uvr_ags)
summary(lm_1uvrr)
resultsvrr = predict(lm_1uvrr, uvr_ags[, c(21, 25, 34,
    35, 42,
    46, 52)], type = "response")
resultsvrr = ifelse(resultsvrr > 0.5, 'W', 'L')
table(uvr_ags$Result, resultsvrr)

# Creates revised logistic regression model and assesses the accuracy of the
# model. Ranked vs. unranked.
lm_1rvur = glm(formula = Result ~ Game + Conference + Pass.Att.For +
    Pass.Yds.For + Pass.TD.For + Rush.Yds.For + Rush.TD.For +
    Penalty.FD.For + Fumbles.For + Int.For + Pass.Cmp.Against +

```

```

    Pass.Att.Against + Pass.Yds.Against + Pass.TD.Against +
    Rush.Yds.Against + Rush.TD.Against + Penalties.Against +
    Fumbles.Against + Int.Against,
    family = "binomial", data = rvu_ags)
resultsrvu = predict(lm_1rvur, rvu_ags[, c(2, 8, 18, 20, 21, 23,
                                         25, 31, 35, 36, 38, 39, 41, 42, 44,
                                         46, 54, 56, 57)], type = "response")
resultsrvu = ifelse(resultsrvu > 0.5, 'W', 'L')
table(rvu_ags$Result, resultsrvu)

# Creates revised logistic regression model and assesses the accuracy of the
# model. Unranked vs. unranked.
lm_1uvur = glm(formula = Result ~ Site + Pass.Att.For + Pass.Yds.For +
               Pass.TD.For + Rush.TD.For + Pass.FD.For + Rush.FD.For +
               Penalty.FD.For + Fumbles.For + Int.For + Pass.Cmp.Against +
               Pass.Att.Against + Pass.Yds.Against + Pass.TD.Against +
               Rush.Att.Against + Rush.Yds.Against + Rush.TD.Against +
               Pass.FD.Against + Rush.FD.Against + Penalty.FD.Against +
               Penalties.Against + Penalty.Yds.Against +
               Fumbles.Against + Int.Against,
               family = "binomial", data = uvu_ags)

# Looks at correlations.
corrplot(cor(uvu_ags[,c(17, 19, 20, 24, 28, 29, 30, 34,
                       35, 37, 38, 40, 41, 42, 43, 45, 49, 50,
                       51, 53, 54, 55, 56)]))

# Redo model with predictors with high coefficients (above 500).
lm_1uvur = glm(formula = Result ~ Site +
               Pass.TD.For + Rush.TD.For + Pass.FD.For + Rush.FD.For +
               Penalty.FD.For + Fumbles.For + Int.For + Pass.TD.Against +
               Rush.TD.Against + Pass.FD.Against + Rush.FD.Against +
               Penalty.FD.Against + Penalties.Against +
               Fumbles.Against + Int.Against,
               family = "binomial", data = uvu_ags)
resultsvuvu = predict(lm_1uvur, uvu_ags[, c(6, 20, 24, 28, 29, 30, 34,
                                             35, 41, 45, 49, 50,
                                             51, 53, 55, 56)], type = "response")
resultsvuvu = ifelse(resultsvuvu > 0.5, 'W', 'L')
table(uvu_ags$Result, resultsvuvu)

# Creates logistic regression models to determine weights for each of the games
# in the dataset.
# None exists for the unranked vs. unranked category--weight is set at 0.1.
lm_2rvr = glm(Result ~ TeamRank + OppRank, data = rvr_ags, family = "binomial")
lm_2uvr = glm(Result ~ OppRank, data = uvr_ags, family = "binomial")
lm_2rvu = glm(Result ~ TeamRank, data = rvu_ags, family = "binomial")

## ADJACENCY MATRICES

# Creates list of all unique teams from the main dataset.
unique_teams = unique(c(levels(all_game_stats[, 5]), levels(all_game_stats[, 7])))

# Creates adjacency matrix for each year to house the scores that will be used to
# rank the top 25 teams in order.
strength_2014 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))
colnames(strength_2014) = unique_teams; rownames(strength_2014) = unique_teams

strength_2015 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))
colnames(strength_2015) = unique_teams; rownames(strength_2015) = unique_teams

strength_2016 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))
colnames(strength_2016) = unique_teams; rownames(strength_2016) = unique_teams

strength_2017 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))

```

```

colnames(strength_2017) = unique_teams; rownames(strength_2017) = unique_teams

strength_2018 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))
colnames(strength_2018) = unique_teams; rownames(strength_2018) = unique_teams

strength_2019 = matrix(0, nrow = length(unique_teams), ncol = length(unique_teams))
colnames(strength_2019) = unique_teams; rownames(strength_2019) = unique_teams

## PREDICTIONS

# Predicts the probability, weight, and ultimately the score of each game in the
# ranked vs. ranked category. For each section in the if-else statement, the first
# line predicts the score, the second line predicts the weight, the third line
# creates the transformation of the probability, the fourth line creates the
# transformation of the weight, and the fifth line creates the score for the
# specific game in the adjacency matrix.
for (game in 1:447) {
  if (game < 88){
    prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
                                                    36, 37, 40, 43, 44, 45, 47, 51,
                                                    57, 58)], type = "response")
    prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
    prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
    prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
    strength_2014[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =
      strength_2014[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
      prob_allrvr * prob_rankrvr
  }else if (game < 164){
    prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
                                                    36, 37, 40, 43, 44, 45, 47, 51,
                                                    57, 58)], type = "response")
    prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
    prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
    prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
    strength_2015[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =
      strength_2015[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
      prob_allrvr * prob_rankrvr
  }else if (game < 228){
    prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
                                                    36, 37, 40, 43, 44, 45, 47, 51,
                                                    57, 58)], type = "response")
    prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
    prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
    prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
    strength_2016[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =
      strength_2016[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
      prob_allrvr * prob_rankrvr
  }else if (game < 305){
    prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
                                                    36, 37, 40, 43, 44, 45, 47, 51,
                                                    57, 58)], type = "response")
    prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
    prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
    prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
    strength_2017[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =
      strength_2017[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
      prob_allrvr * prob_rankrvr
  }else if (game < 372){
    prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
                                                    36, 37, 40, 43, 44, 45, 47, 51,
                                                    57, 58)], type = "response")
    prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
    prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
    prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
    strength_2018[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =

```

```

    strength_2018[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
    prob_allrvr * prob_rankrvr
} else {
  prob_allrvr = predict(lm_1rvrr, rvr_ags[game, c(19, 22, 23, 24, 26, 30,
    36, 37, 40, 43, 44, 45, 47, 51,
    57, 58)], type = "response")
  prob_rankrvr = predict(lm_2rvr, rvr_ags[game, c(15, 16)], type = "response")
  prob_allrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_allrvr, -prob_allrvr)
  prob_rankrvr = ifelse(rvr_ags[game, 9] == "W", 1 - prob_rankrvr, prob_rankrvr)
  strength_2019[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] =
  strength_2019[as.character(rvr_ags[game, 5]), as.character(rvr_ags[game, 7])] +
  prob_allrvr * prob_rankrvr
}
}

# Predicts the probability, weight, and ultimately the score of each game in the
# unranked vs. ranked category. For each section in the if-else statement, the
# first line predicts the score, the second line predicts the weight, the third
# line creates the transformation of the probability, the fourth line creates the
# transformation of the weight, and the fifth line creates the score for the
# specific game in the adjacency matrix.
for (game in 1:96){
  if (game < 11) {
    prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
    35, 42,
    46, 52)], type = "response")
    prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
    prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
    prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
    strength_2014[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
    strength_2014[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
    prob_alluvr * prob_rankuvr
  } else if (game < 24) {
    prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
    35, 42,
    46, 52)], type = "response")
    prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
    prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
    prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
    strength_2015[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
    strength_2015[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
    prob_alluvr * prob_rankuvr
  } else if (game < 44) {
    prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
    35, 42,
    46, 52)], type = "response")
    prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
    prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
    prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
    strength_2016[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
    strength_2016[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
    prob_alluvr * prob_rankuvr
  } else if (game < 59) {
    prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
    35, 42,
    46, 52)], type = "response")
    prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
    prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
    prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
    strength_2017[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
    strength_2017[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
    prob_alluvr * prob_rankuvr
  } else if (game < 85) {
    prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
    35, 42,

```

```

                                46, 52)], type = "response")
prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
strength_2018[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
  strength_2018[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
  prob_alluvr * prob_rankuvr
}else {
  prob_alluvr = predict(lm_1uvrr, uvr_ags[game, c(21, 25, 34,
                                35, 42,
                                46, 52)], type = "response")
  prob_rankuvr = predict(lm_2uvr, uvr_ags[game, c(15, 16)], type = "response")
  prob_alluvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_alluvr, -prob_alluvr)
  prob_rankuvr = ifelse(uvr_ags[game, 9] == "W", 1 - prob_rankuvr, prob_rankuvr)
  strength_2019[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] =
    strength_2019[as.character(uvr_ags[game, 5]), as.character(uvr_ags[game, 7])] +
    prob_alluvr * prob_rankuvr
}
}

# Predicts the probability, weight, and ultimately the score of each game in the
# ranked vs. unranked category. For each section in the if-else statement, the
# first line predicts the score, the second line predicts the weight, the third
# line creates the transformation of the probability, the fourth line creates the
# transformation of the weight, and the fifth line creates the score for the
# specific game in the adjacency matrix.
for (game in 1:937){
  if (game < 145) {
    prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
                                                  36, 38, 39, 41, 42, 44, 46, 54,
                                                  56, 57)], type = "response")
    prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
    prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
    prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
    strength_2014[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =
      strength_2014[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
      prob_allrvu * prob_rankrvu
  }else if (game < 304) {
    prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
                                                  36, 38, 39, 41, 42, 44, 46, 54,
                                                  56, 57)], type = "response")
    prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
    prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
    prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
    strength_2015[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =
      strength_2015[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
      prob_allrvu * prob_rankrvu
  }else if (game < 444) {
    prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
                                                  36, 38, 39, 41, 42, 44, 46, 54,
                                                  56, 57)], type = "response")
    prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
    prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
    prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
    strength_2016[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =
      strength_2016[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
      prob_allrvu * prob_rankrvu
  }else if (game < 614) {
    prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
                                                  36, 38, 39, 41, 42, 44, 46, 54,
                                                  56, 57)], type = "response")
    prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
    prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
    prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
    strength_2017[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =

```

```

    strength_2017[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
    prob_allrvu * prob_rankrvu
}else if (game < 766) {
  prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
    36, 38, 39, 41, 42, 44, 46, 54,
    56, 57)], type = "response")
  prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
  prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
  prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
  strength_2018[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =
    strength_2018[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
    prob_allrvu * prob_rankrvu
}else {
  prob_allrvu = predict(lm_1rvur, rvu_ags[game, c(2, 8, 18, 20, 21, 23, 25, 31, 35,
    36, 38, 39, 41, 42, 44, 46, 54,
    56, 57)], type = "response")
  prob_rankrvu = predict(lm_2rvu, rvu_ags[game, c(15, 16)], type = "response")
  prob_allrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_allrvu, -prob_allrvu)
  prob_rankrvu = ifelse(rvu_ags[game, 9] == "W", 1 - prob_rankrvu, prob_rankrvu)
  strength_2019[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] =
    strength_2019[as.character(rvu_ags[game, 5]), as.character(rvu_ags[game, 7])] +
    prob_allrvu * prob_rankrvu
}
}

# Predicts the probability, weight, and ultimately the score of each game in the
# unranked vs. unranked category. For each section in the if-else statement, the
# first line predicts the score, the second line creates the transformation of the
# probability, and the third line creates the score for the specific game in the
# adjacency matrix using the score and the weight of 0.1.
for (game in 1:381){
  if (game < 69) {
    prob_alluvu = predict(lm_1uvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,
    35, 41, 45, 49, 50,
    51, 53, 55, 56)], type = "response")
    prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
    strength_2014[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
      strength_2014[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
      prob_alluvu * UNRANKED_WEIGHT
  }else if (game < 130) {
    prob_alluvu = predict(lm_1uvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,
    35, 41, 45, 49, 50,
    51, 53, 55, 56)], type = "response")
    prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
    strength_2015[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
      strength_2015[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
      prob_alluvu * UNRANKED_WEIGHT
  }else if (game < 217) {
    prob_alluvu = predict(lm_1uvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,
    35, 41, 45, 49, 50,
    51, 53, 55, 56)], type = "response")
    prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
    strength_2016[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
      strength_2016[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
      prob_alluvu * UNRANKED_WEIGHT
  }else if (game < 266) {
    prob_alluvu = predict(lm_1uvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,
    35, 41, 45, 49, 50,
    51, 53, 55, 56)], type = "response")
    prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
    strength_2017[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
      strength_2017[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
      prob_alluvu * UNRANKED_WEIGHT
  }else if (game < 331) {
    prob_alluvu = predict(lm_1uvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,

```

```

                                35, 41, 45, 49, 50,
                                51, 53, 55, 56)], type = "response")
prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
strength_2018[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
  strength_2018[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
  prob_alluvu * UNRANKED_WEIGHT
} else {
  prob_alluvu = predict(lm_luvur, uvu_ags[game, c(6, 20, 24, 28, 29, 30, 34,
                                35, 41, 45, 49, 50,
                                51, 53, 55, 56)], type = "response")
  prob_alluvu = ifelse(uvu_ags[game, 9] == "W", 1 - prob_alluvu, -prob_alluvu)
  strength_2019[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] =
    strength_2019[as.character(uvu_ags[game, 5]), as.character(uvu_ags[game, 7])] +
    prob_alluvu * UNRANKED_WEIGHT
}
}
}

## CREATING RANKINGS

# The following organizes the rankings based off the adjacency matrices.
first = rowSums(strength_2014)[order(rowSums(strength_2014))]
first_write = rev(first[which(names(first) %in%
                             unique(all_game_stats$Team[all_game_stats$X < 337]))])
second = rowSums(strength_2015)[order(rowSums(strength_2015))]
second_write = rev(second[which(names(second) %in%
                                 unique(all_game_stats$Team[which(all_game_stats$X > 336 &
                                                                    all_game_stats$X < 673))])])
third = rowSums(strength_2016)[order(rowSums(strength_2016))]
third_write = rev(third[which(names(third) %in%
                               unique(all_game_stats$Team[which(all_game_stats$X > 672 &
                                                                    all_game_stats$X < 1009))])])
fourth = rowSums(strength_2017)[order(rowSums(strength_2017))]
fourth_write = rev(fourth[which(names(fourth) %in%
                                 unique(all_game_stats$Team[which(all_game_stats$X > 1008 &
                                                                    all_game_stats$X < 1346))])])
fifth = rowSums(strength_2018)[order(rowSums(strength_2018))]
fifth_write = rev(fifth[which(names(fifth) %in%
                               unique(all_game_stats$Team[which(all_game_stats$X > 1345 &
                                                                    all_game_stats$X < 1682))])])
sixth = rowSums(strength_2019)[order(rowSums(strength_2019))]
sixth_write = rev(sixth[which(names(sixth) %in%
                               unique(all_game_stats$Team[which(all_game_stats$X > 1681 &
                                                                    all_game_stats$X < 2023))])])

# Writes our rankings to .csv files.
write.csv(first_write, "2014_predictions3.csv")
write.csv(second_write, "2015_predictions3.csv")
write.csv(third_write, "2016_predictions3.csv")
write.csv(fourth_write, "2017_predictions3.csv")
write.csv(fifth_write, "2018_predictions3.csv")
write.csv(sixth_write, "2019_predictions3.csv")

#####

```

(J. Lawson) THE UNIVERSITY OF ARIZONA, TUCSON, AZ 85721, USA

E-mail address, J. Lawson: jerlawson13@email.arizona.edu, jboloslawson@gmail.com