COSC 445: College Football Data Modeling

Kellen Leland

I. OBJECTIVE

The objective of this project is to collect various college football data, clean the data, and use it to create statistical models that will rank teams and predict weekly matchups. Prediction of weekly matchups will be separated into different categories: predicting which team will win the game, predicting the matchup against the spread, and predicting the matchup point total (over/under). Each of these models will have variations that can be compared and an analysis of each of the models' success will be completed at the end of the project.

I have always enjoyed watching and following college football. My father and I have a friendly competition each college football season. We each pick 10 games a week, and whomever has the highest percentage of accuracy at the end of the season is crowned as the winner.

In the past, I have always made my picks based on gut feelings and general knowledge of the various teams which has led to varying levels of success. I have recently become more interested in the data analytics that go into these predictions as there are vast amounts of raw and compiled data available to the public for free.

I aim to improve the accuracy of my weekly picks by using digital archaeology techniques and data analytics to build models that make the predictions in favor of my usual very subjective human analysis.

II. DATA

All of my data is pulled from a free and public college football data API. The API's web interface is located at api.collegefootballdata.com. You can actually request and download data into JSON or CSV formats directly on the website. You can also use the python requests library or one of several custom python wrapper modules to request the data. This API has a wealth of college football data points and metrics, and allows unlimited requests to users for free. My models and analysis focus on a handful of data points and metrics that I thought would bring the best results, but there are hundreds more I could have

added into the models' analysis. There are seemingly unlimited options and combinations of data that could be played with and tested when creating and fine tuning predictive models for college football, and this is all from one source. There are many other sources out there with different data and metrics as well. I would eventually like to add in the ability to scrape some custom and specialty metrics from specific websites and curators that have proven success in building accurate football prediction models.

The goal of my models is to be able to more accurately predict which team will 'cover the spread' and if the point total will go over or under the set line total compared to using a purely subjective analysis. The spread can be thought of as a way to even the odds between two unevenly matched teams. If the spread is negative, that team must win by a margin greater than the negative number. This can be thought of as a handicap for the team that is favored. If a spread is positive, the team can lose by that number of points or less and still 'cover the spread' and win the matchup. The total is easier to understand, you are picking whether or not you believe the total number of points in the game will go over or under the set value. The model will obviously also state who it predicts will win the game straight up. In the betting world this is called the 'moneyline', but my models and analysis will be focused on the spreads and totals.

Since all of my data comes from this API, there was minimal cleaning involved. The only things I really needed to do to prepare the data was make all the data relative to the home team in order to make the calculations simpler to transfer into python code, and to put them into some custom data structures to make the analysis easier.

III. MODEL BUILDING

The first model I built to predict the weekly college football matchups is based on a Monte Carlo Simulation. A Monte Carlo simulation is a mathematical method for calculating the odds of multiple possible outcomes occurring in an uncertain process through repeated random sampling. This type of simulation was developed by John Von Neumann and Stanislaw Ulam during World War II as a method for military leaders to make better and more logically informed decisions. The Monte Carlo simulation works by running a large number of simulations of each game and as N becomes very large, the algorithm converges on the most likely solution in the solution space. The inputs I use for this simulation are the

average points scored, the average points allowed, the adjusted points scored, the adjusted points allowed, the standard deviation of the points adjusted points scored, and a pseudo-random number between zero and one. These values are used in an inverse normal cumulative distribution function which outputs the simulated score for each team. This is run 10,000 times per game and the outputs are used to pick which side of the spread and which side of the over/under is the most probable to be correct. I also added the ability to use other metrics as modifiers. These include home field advantage, talent rankings, explosiveness ranking, red zone efficiency, and form. These metrics are normalized and applied to the scores after the above transformation as multipliers.

I then decided to take another approach for a different predictive model. The second model is an artificial neural network that is trained on 5 years of college football data. The neural network is set up to output the predicted margin and total for each game. The features used during the training of the network is the greatest common subset of data points and metrics pulled from the API for all games from 2015-2020. I used a python module called fast.ai which runs off PyTorch to create, train, and run the artificial neural network. The network outputs into a numpy array, which is then converted into a pandas dataframe and exported as a CSV file so the win and loss percentages can be calculated in an excel spreadsheet.

I also created a tool that tracks line movement for every game each week. Sportsbooks and casinos create and actively adjust lines so that either side has an equal amount of bettors. This means as people place bets, the lines are changed to try to keep that balance. This tool allows the user to see what the general (betting) public thinks about a matchup. This tool is especially helpful if the user has access to a locked line, allowing them to find values in lines that have since changed. It can also help clue the user in to possible extraneous circumstances like inclement weather or injuries to players of high significance.

IV. RESULTS

The Monte Carlo simulation model gave me the best results, with a 50.33% spread win percentage and a 62.92% over/under win percentage for the 2021 season. There were weeks with win percentages as high as 82%. For comparison, my average win percentage for the 2021 season using strictly subjective analysis was 52%. I found that the modifiers did not change the overall win percentages

in a significant way, but would like to continue testing the addition of other metrics to see if these numbers can be improved.

The artificial neural network model had slightly worse results, with a win percentage of 48% for both the spread and the over/under. This model is harder to fine tune, as the abstraction involved in an artificial neural network module means it is near impossible to know how the model is weighting the different features it uses during training and prediction. I do not believe that this approach is the best for this kind of use case given the dynamic nature of college football and how teams change season to season. I think limiting the training data to one or two seasons prior, and limiting the set of data points and metrics fed in as features could help improve its accuracy.

After creating these models and tools, I believe the best approach to picking spreads and totals is a multifaceted one involving analyzing the output of multiple models and using other tools to help find outlier games with the most value. These outlier games would be the ones with the greatest difference in spread/total and the predicted margin of victory and total points.

V. CHALLENGES

The main challenge I came across during this project is just figuring out where to start with the mathematics involved in creating a predictive football model. People, both online and offline, are very protective of their models and sparsely share details of their inner-workings. Once I figured out how to create each of the models it was difficult to find a way to add more variables, especially with the Monte Carlo simulations. I am not sure the approach I took is the best one, and I think there's probably a way to add complexity directly into the inverse cumulative distribution function itself rather than normalizing the modifiers and applying them after the fact. Another difficulty was deciding which metrics matter most when predicting a football game. This is something that requires time, patience, and fine tuning over the course of many seasons in order to get a more consistent result. I also think, and my results show, that these types of models tend to be better at predicting the total number of points in a game than which side of the spread will cover.

VI. FUTURE WORK

I would like to add functionality to the line movement tool so it also shows the movement in the total for each game. I would like to spend more time fine tuning the inputs for the Monte Carlo simulation, and possibly find a different way of adding more complexity before the transformation function. Training the artificial neural network model on less and more recent data points could also make it more accurate. I plan to add the ability to scrape other custom and specialty metrics that are only available on certain websites. These metrics are valuable and have a track record of success in predictive modeling for college football. Lastly, I would like to create a tool that can take the output from both models and highlight outlier games that have the most value to be exploited.