College Football Bowl Game Predictor

Evan Cheshire **Austin Childs** **Thomas Leung**

1 Introduction

College football is one of many great traditions found in American universities. The football team provides a source of entertainment and a sense of community for both students and alumni, and many fans spend three hours every Saturday in the fall rooting for their team. The season is quick, and broken into two parts, a regular season and a playoff. The regular season is composed on average of about 12 games per team, most within the same conference. At the end of the season, many teams with winning records are invited to a bowl game. The bowl game is the final game of the season for every team and is the best part of the season for many college football fans, as teams who appear to be evenly matched play each other in many exciting games.

For many fans, bowl season leads to gambling. Specifically, out of the list of many bowl games, fans pick winners of each separate bowl. They also assign a confidence that their pick is correct, ranging from 1-#of bowl games. The number that is assigned can only be used once. If the fan picks the winner of the game, then he gets the total points assigned for that bowl game. At the end of bowl season, the person with the maximum score wins. A combination of picking winners in bowl games and placing high confidence on bowl games in necessary for winning the game.

The goal of our project is to maximize the potential score of this described game. Based on the regular season statistics of both teams within the matchup of the bowl game, we calculate a winner and a potential difference in score. The greater the score difference between individual teams, the more likely the win. Thus from the predicted score difference, we would have a winner and a confidence measurement for each bowl game.

2 Related Work

Previous Stanford projects have focused on predicting the winners of bowl games based on regular season statistics, predicting upsets in college football, and predicting the winners of NFL games. In general, the past projects were classification problems and used methods including logistic regression, Naive Bayes, and Support Vector Machines [1, 2, 3, 4]. For comparison, we looked at the results of NFL game prediction models such as Microsoft's Cortana model, accurate to 67%, and Nate Silver's Elo model, accurate to 70% [5, 6]. On the other hand, the goal of our project is to also assign a confidence to the predicted outcome of each game, so our goal is regression of either the scores or the differential.

3 Data

For our training and testing data, we did not need to scrape the data as we found the data on a blog [7]. The data from 2000-2010 was reported for all variables that we wished to test. We created a corresponding MySQL database, and were able to use Python and MATLAB to query this data. We averaged the regular season game statistics for each team, which we used as features for our models. The data contained approximately 350 bowl games, and 36 features per team. The features were averages or variances of the offense and defense over their regular season performance.

4 Model Specifications

We took two approaches during our initial modeling stage, both utilizing linear regression. The first was to predict what a team would score given their average offensive statistics over the regular season and their opponent's defensive statistics over the same time period. As an alternative, we built another model to predict a margin of victory given all of the information about both teams in the bowl game.

4.1 Score Regression and SVMs

The first began by running linear regression in order to get the projected scores and from that information calculate a spread between the two teams. The higher the spread, the more confidence we have in the pick. As a baseline model, we included all of the features we had available for the regression. For the test set, we parsed out a single season data and used the remaining ten as training data. This was iterated over every season, thus giving us 11-Fold Cross Validation. When predicting scores, we had an average MSE (across all 11 years) of 174.75. To follow this up we used the predicted scores to calculate which team would win the game and had an average MSE of 0.4857. This provides very little lift over simply using a coin to randomly pick winners.

In an attempt to improve our results and better the model, we implemented Best Subset, Forward Stepwise Selection, and Backward Stepwise Selection for feature selection. These resulted in two subsets of variables to use for predicting score and winning team respectively as seen in Appendix: Table A. Regarding the score predictions, our average MSE was 150.20 and forecasting the victor had an average MSE of 0.4408. Thus we had marginally better results, but still not acceptable

Following feature selection, we attempted to utilize an SVM to both regress the score and classify the winner. Using all features left us with respective average MSEs of 161.99 and 0.4497 respectively. Without tuning, this method performed essentially the same as the linear regression with specific parameters.

Finally, we used a logistic model to predict the probability of each team winning and then selected the team with the highest odds. We ran a baseline model with every parameter and then used the same feature selection methods as before (best subset and stepwise) to narrow them down. This resulted in an MSE of 0.4832 for baseline and 0.4347 for the reduced model. Clearly, the logistic regression was most successful but still not satisfactory. The winner selection accuracy across all seasons can be seen in Figure 1.

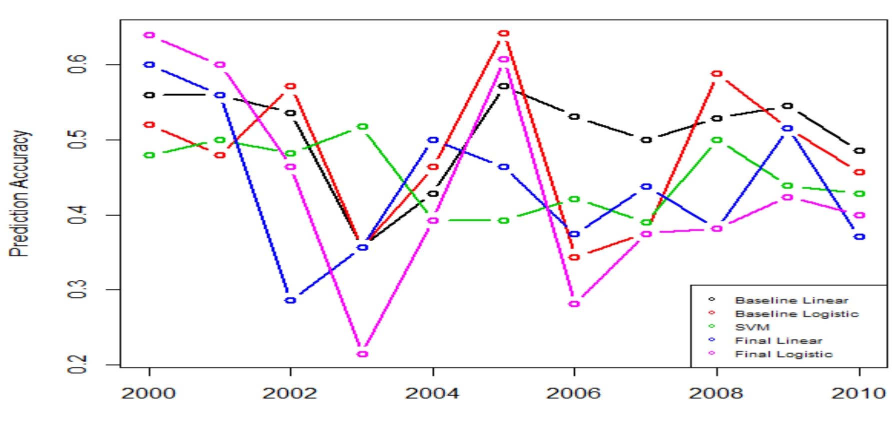

Figure 1: Prediction Accuracy for Score Models

4.2 Differential Regression

The second model was a linear regression on score differential. Instead of predicting score based on one team's offense and the opposing team's defense, a score difference was projected from both offensive and defensive factors from both teams. Like the original score model, we used a cross-validation approach on a season by season projection to assess the model. Again, one season would be the test set, and all other season would be considered training data. The initial model used every single variable from both teams offense and defense. This was to establish an initial baseline upon which to improve and compare. The results of this model were less than perfect. The average came to be around a coin flip, which is pretty inadequate for selecting a winner. This is the "Baseline" plot in Figure 2.

On top of this initial model, we ran a feature selection to find out which features truly contributed to the overall model. Essentially, each feature had a p-value, a measurement of how much the feature was contributing to the overall data set. From 2000-2010, 11 different regressions could be had, with one season left out each time. The p-values would be different for the parameters with separate years left out. The initial rule was established that for all 11 models, a feature would remain if the min(p-values) < 0.1 and the max(p-values) < 0.4.

With this initial rule, a jump in performance was noticed immediately. However, certain statistics would appear for one team, but not for the other team, i.e. Team 1 Offense Average Points would be in these statistics, while Team 2 Offense Average Points would not be in the set. Thus we weren't necessarily confident that these features could describe our model correctly.

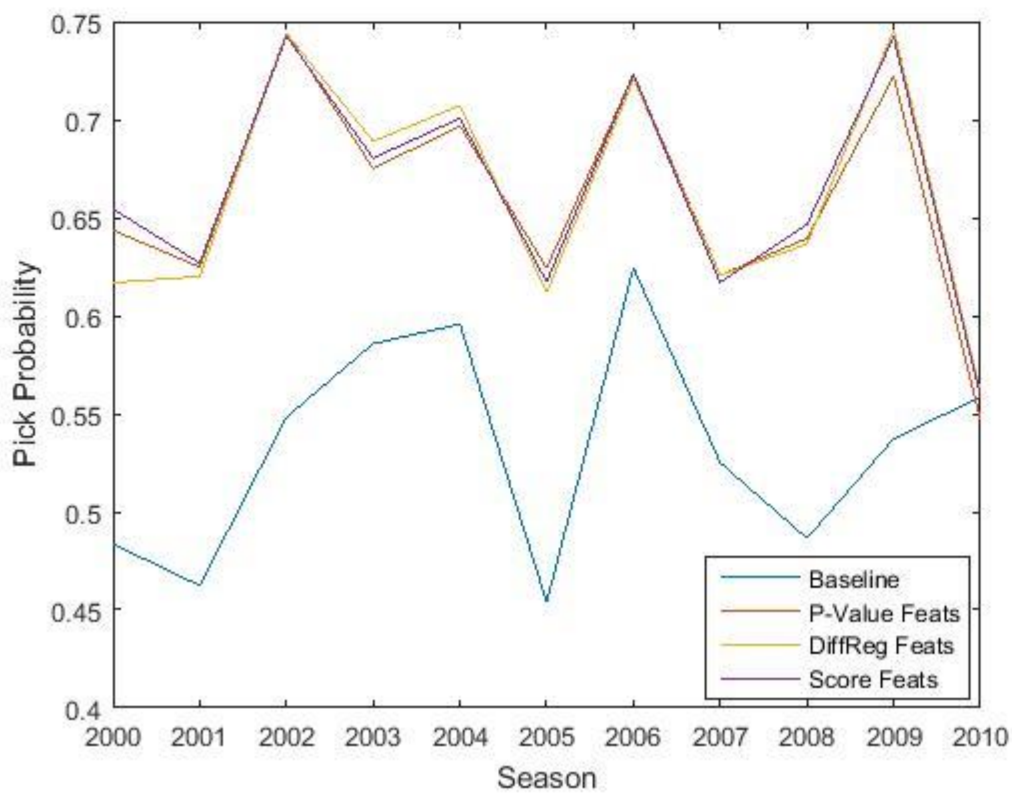

Figure 2: Prediction Accuracy for Differential Models

We were able to hone in on variables with randomization. Unlike the score linear regression, where the data is set, we could randomize the matchup so that the teams in the Team 1 and Team 2 column would be different for every regression. Running this over 100 randomizations, we saved the features in which the p-values hit the rules, now referred to as α. In Figure 2, the "P-Value Feats" is the average pick percentage where the statistics change for every single iteration of the model based on the p-value rule.

To select finalized features for the differential regression model, we selected the values from α that appeared in at least 20% of the iterations. These values also had to both appear for Team 1 and Team 2. These particular features are in the Appendix table as the differential model. The pick probability on a year by year basis is displayed in Figure 2 as the "DiffReg Feats". The last plot is using the features found in the score regression called "Score Feats". All of these models average around a 66% correct pick probability.

5 Conclusion

Overall, we found the best results using the differential model and running variable selection. The accuracy at 66% is only slightly below professional models, which attain around 69%, and provides a significant lift over the baseline attempts. Predicting individual scores provided little benefit and the methodologies applied to this outcome may be better served in the differential modeling context.

Further improvements could be made given more resources, namely scraping data for different features, running PCA to reduce dimensionality, or developing an anomaly model to try to predict upsets. The last of those suggestions would be the hardest, but also has the highest potential to make significant improvement. Most of the variation between accuracy across seasons is due to the amount of upsets from year to year. Currently, our models are naturally designed to assign predicted victory to the favorite the large majority of the time. A more robust model would incorporate a method to account for upsets.

6 References

[1] B. Hamadani, "Predicting the outcome of NFL games using machine learning." [Online]. Available: http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf.

[2] AS Padron and J. Sinsay, "Upset prediction in college football". [Online]. Available: http://cs229.stanford.edu/proj2013/PadronSinsay-UpsetPredictioninCollegeFootball.pdf

[3] B. Liu and P. Lai, "Beating the NCAA Football Point Spread." [Online]. Available: http://cs229.stanford.edu/proj2010/LiuLai-BeatingTheNCAAFootballPointSpread.pdf

[4] J. Hamann, "What It Takes To Win: A Machine Learning Analysis of the College Football Box Score." [Online]. Available: http://cs229.stanford.edu/proj2011/Hamann-WhatItTakesToWin.pdf

[5] "Cortana's predictions," Dec. 8, 2015. [Online]. Available: http://www.cortanapredictions.com/.

[6] "2015 NFL predictions," Dec. 8, 2015. [Online]. Available: http://projects.fivethirtyeight.com/2015-nfl-predictions/

[7] "Perspectives on College Football," Dec. 8, 2015. [Online]. Available: http://thenationalchampionshipissue.blogspot.com/2005/08/

7 Appendix

Table A: Features for Reduced Models

Feature	Linear Model Score	Linear Model Win	Logistic Model	Differential Score Features	Differential Model
Variance in Defensive Touchdowns Allowed	X	X	X	X	
Defensive Turnovers	X			X	X
Defensive Punt Return Yards Allowed	X		X	X	X
Defensive Field Goal Percentage Allowed	X			X	
Defensive Penalty Yards	X	X	X	X	
Defensive Points Allowed	X			X	
Offensive First Downs	X			X	X
Offensive Kickoff Return Yards	X			X	
Offensive Completion Percentage		X	X		
Defensive Completion Percentage Allowed		X	X		
Defensive Kickoff Return Yards Allowed		X			
Defensive Punt Return Yards Allowed		X	X		
Offensive Penalty Yards					X
Offensive Field Goal Percentage					X
Offensive Punt Return Yards					X
Defensive First Downs Allowed					X