

BAS 474 NFL Project – Tennessee Titans

Nicolas Stevens

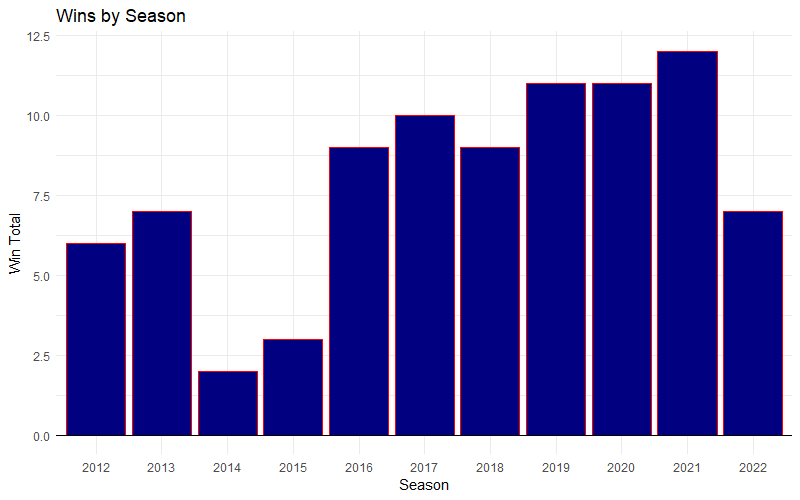
2023-12-6

**Executive Summary**

In this project, I am analyzing the past 10 years of data for the Tennessee Titans. In doing so, I intend to find a model that ultimately leads to more wins for my favorite team (the Titans). The predictor variables I elected to use to generate my models include: Turnover Margin, Third Downs Converted, Third Down Conversions Allowed, number of times Tennessee was sacked during the game, number of times Tennessee sacked the opposing team in a game, Rushing Yards, Passing Attempts, and Number of Opponent Incompletions. I elected to use Point Differential as my outcome / dependent variable. This variable is directly correlated to winning and losing, as a score greater than 0 means a win, and a score less than 0 means a loss. I then ran three different types of predictive models on this data: GLMNet, Gradient Boosting Machine, and Random Forest. After generating the version of each model with the lowest generalization error, I compared them against each other. It was ultimately the GLM net model that had the lowest generalization error out of all of the models by more than one standard deviation. I then utilized the GLM net model to predict the point differential in the hold out data and compared it against the real point differentials. The model ended up having a root mean squared error (RMSE) of 9.398, which was more than 1 standard deviation better than the RMSE of the naïve model (11.955), meaning that the GLM net model does have greater predicting power than a simple average of point differentials over time. I then made suggestions to the Titans based off of the coefficients within the GLM net model. The most important area for the Titans to focus on positively improving is increasing their turnover margin. Meaning they need to throw less interceptions and giving the ball away through fumbles, while getting as many interceptions and fumbles on defense as possible. The area that they need to work on preventing from happening the most is allowing the Titans quarterback from being sacked, as this had the largest negative impact on point differential out of all predictors I tested.

**The Past 10 Seasons**

The Tennessee Titans were quite a bad team from 2012-2015, not having a single winning record over that time span. However, this changed in 2016 with the team’s first winning season since 2011. From 2016 to 2021, the Titans experienced many good years finishing with positive records, and even a deep playoff run in 2019. Below is a bar chart, indicating the team’s number of wins each year from 2012-2022. You can see the clear positive trend in recent years, with an unfortunate decline in 2022.



| **Year** | **Win Total** | **Point Differential** | **Winning Season** |
| --- | --- | --- | --- |
| 2012 | 6 | -141 | No |
| 2013 | 7 | -19 | No |
| 2014 | 2 | -184 | No |
| 2015 | 3 | -124 | No |
| 2016 | 9 | 3 | Yes |
| 2017 | 10 | -42 | Yes |
| 2018 | 9 | 7 | Yes |
| 2019 | 11 | 83 | Yes |
| 2020 | 11 | 45 | Yes |
| 2021 | 12 | 62 | Yes |
| 2022 | 7 | -61 | No |

In the bar chart below, you can see the team’s point differential over the years. Virtually all winning seasons finish above 0 in point differential, with the exception of 2017. A season like 2017 is typically indicative of losing by large margins and winning tight games. However, on the game-by-game level, point differential does not have exceptions. If you a team scores more than the other, they win. I have also provided a table with annual win totals by year, and point differential, as well as if the Titans finished with a “Winning Season” meaning they had a record of greater than .500, which in the NFL is 9 or more wins.

A graph with blue rectangular bars

Description automatically generated with medium confidence

**Creating the Holdout Set**

For my analysis, I have elected to use the seasons from 2012-2021 as my training set. A training set is the set of data that the predictive model will “train” itself on. Through the process of K-Fold Cross validation, the model will come up with the best model that results in the lowest generalization error given the prescribed data and predictors. Models with lower generalization error make the most accurate predictions in the long run on new individuals that the model has not yet seen.

2022 will be my holdout set and will be what I use to see how my model’s predictions compare to reality. The holdout set is our sanity check to see if the model generated through the training procedure does a good job of predicting based on data it has not yet seen before. We then look at the generalization error resulting from our predictions on the holdout data to see which type of model has done the best job in predicting our variable of interest, which in this case is point differential.

**Model Creation**

The 3 models I decided to create were a GLM net model, a random forest model, and a Gradient Boosting Machine Model. As mentioned earlier I utilized a variety of variables, which I will now explain in more detail. Turnover margin was calculated by adding all of the turnovers Tennessee was able to force against its opponents (meaning both fumbles forced and recovered, as well as interceptions), and subtracting all of the turnovers that Tennessee gave away to opposing teams in an individual game. Thirds down converted is the total number of times that the Titans were able to get a first down while being on third down. Third downs allowed is the number of times that the Titans allowed opponents to do convert a first down on third down. Another variable utilized was simply the number of times that on offensive play for Tennessee resulted in a sack in a single game. I also decided to utilize the number of times Tennessee was able to sack the opposing team in a game as a predictor variable. Rushing yards is the total yardage accumulated during run plays for the Titans in a single game. Passing attempts is the total number of times the Titans attempted to pass the ball in a single game. Finally, I utilized the number of incompletions from opposing teams in a single game as a predictor variable. My outcome variable was point differential, which is a very good numeric analog for winning and losing, as a negative value is always a loss, and a positive value is always a win.

The first model I used was GLM net. GLM net is a form of a regularized linear regression model. Regularization helps to eliminate an important issue in non-regularized linear regression models, only minimizing bias and ignoring variance. It is very important for models to have both low variance and low bias simultaneously, at least to the greatest degree of which that is possible. In a standard linear regression model, the goal is simply to minimize the sum of squared errors (SSE). The sum of squared errors is the sum of all of the distances each individual real data point has from the regression line. Regularization takes into account variance by adding a “shrinkage penalty” when calculating it’s SSE. This penalty helps to ensure that coefficients are as small as possible, which ultimately results in both lower variance within the model. Two common forms of regularization are Ridge Regressions and Lasso Regressions, these both have their own strengths and weaknesses. However, GLM net is essentially a combination of both of these regularized models that helps to get the best of both worlds. Because of these factors, I felt that it was a great model to utilize within my testing.

The next model I used was the random forest model. The random forest is essentially a set of large decision trees built on bootstrapped training sets. Let’s break that description down into its parts. A decision tree is model that splits a starting “node” into two smaller nodes of data based off of some rule (x < 5 for example, where x is a predictor) . The rule is determined based on what rule will decrease the sum of squared errors the most from one split, and this continues on based on how many splits you want the tree to have. The predicted y-value for each node is simply the average y-value of each node. These models are typically high variance

and low bias. Random forest models take many of these decision trees, that are independent of one another (meaning they are uncorrelated) and averages their predictions. In order to create lots of extra data sets for decision trees to be ran on, the bootstrapping process is utilized. Bootstrapping is simply taking a random sample with the same size of the original data, from the original data. However, the data is allowed to be selected with replacement meaning a dataset containing only the values 1,2, and 3, can result in a data set of 1,1,1 or 2,3,2 etc. Once the multiple decisions trees have been generated from the bootstrapped data, and their predictions average out, it results in a lower variance model. This makes the random forest model another very enticing option for the modeling of this data.

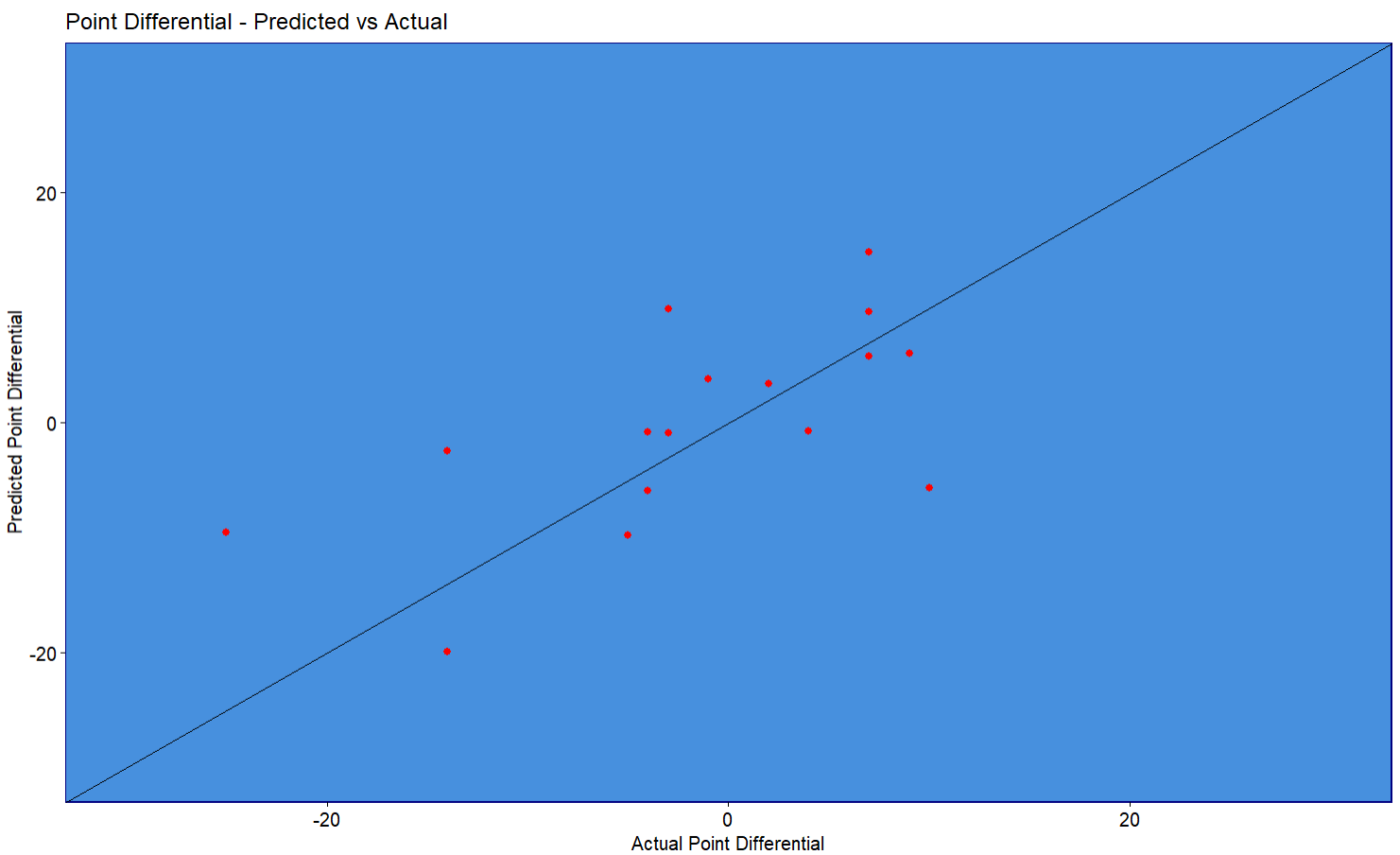
| **Model** | **RMSE** | **RMSE SD** |
| --- | --- | --- |
| GLMnet | 8.53309 | 1.7676120 |
| Random Forest | 10.04826 | 0.2432264 |
| GBM | 10.07559 | 0.6188943 |

The last model predictive model I elected to create was the Gradient Boosting Machine model (GBM). The Gradient Boosting Machine model is similar to the random forest model, in that it relies upon multiple decision trees to come up with a final model. The difference with the GBM is that it does not utilize bootstrapping. Instead it relies upon a number of simpler decision trees (1-2 splits), that aim to improve upon the results of the previous tree before it. The idea is simply that taking a weighted average of the predictions of high bias models can actually reduce the overall bias, resulting in a more accurate model that has both a lower bias and variance.

The GLM net model had the lowest Root Mean Squared Error out of all of the models I tested at around 8.53. Root Mean Squared Error (RMSE) is simply the typical amount by which the model misses, which in this case would be in the units of points in the point differential. The RMSE Standard Deviation was about 1.78, which was the highest for the models I tested. This is the typical difference between the RMSE of a model that has been computed multiple times through the K-Fold Cross Validation Process. The Random Forest and Gradient Boosting Machine models resulted in very similar values of 10.048 and 10.076 respectively. The RMSE SD for the Random Forest model was considerably lower than that of the Gradient Boosting machine with values of 0.243 and 0.619 respectively. Utilizing the 1 standard deviation rule, I was able to conclude that the GLM Net model was the most effective, as it had a lower RMSE than both models by more than 1 of the largest standard deviations.

**Holdout Predictions**

Upon concluding that the GLM Net Model was the best out of all those I tested, I moved on to utilizing the holdout data, and compared predictions of the model to reality. The GLM Net model ended up with a RMSE of 9.398 when used on the holdout data. It is not a surprise that it’s error was higher than it was for the training data. Compared to the naïve model, which is simply the average point differential for the holdout data, it did better. The naïve model had an RMSE of 11.955, which is more than 1 standard deviation larger than that of the GLM net model. Below is a plot with the predicted point differential on the Y-axis and the real point differential on the X-axis. The closer the point is to the line, the closer the prediction was to the actual point differential for that specific game.



**Suggestions to the Team**

I would suggest to the Tennessee Titans that this model does have predictive power. It is not the strong predictive model in the world to be sure, but there are things to be learned from this model that can impact games. Firstly, I would strongly recommend on focusing on maximizing your turnover margin. It had the largest positive regression coefficient within the GLM net model. Additionally, do your absolute best to protect the quarterback, out of the predictors I tested, this had the most negative regression coefficient, meaning that the more times the quarterback is sacked, the lower point differential goes. Alternatively, sacking the opposing team’s quarterback has a very similar regression coefficient, but positive. Third, I would say that Tennessee should continue to look to run the ball, it had the 2nd highest positive regression coefficient in the model and has been a focus of this team throughout its history. There are suggestions that can be made from this model, but I would say based off of size of the regression coefficients, these are the most important.

It's also important to understand that this model is wrong a lot of the time, but it can be a useful tool to help demonstrate areas of importance for the team. The larger the coefficient, the more impactful on the point differential, which ultimately is impactful on winning and losing. As far as how these suggestions can be put into practice, the team should look toward devoting resources towards the offensive line, as this helps to prevent the Titan’s quarterback from being sacked. Additionally, this suggestion should help turnover margin as a better offensive line allows for the quarterback to not get strip sacked in the pocket resulting in fumbles, as well as more time to make unrushed decisions when passing the ball, theoretically leading to less interceptions.