Using Supervised Learning to Predict Football Wins

Arjun Ganesan Daniel Ngo
University of Oklahoma
CS 5033 – Spring 2022

Abstract

American football is a sport with a complex ruleset, but there is some relationship between a team's offensive and defensive performance and wins. In this report, we discuss the hypotheses, experiments, and results of our research. We propose to create models using supervised machine learning that can output how many regular season wins an NFL team can expect to win given a selection certain team statistics.

Project Domain

The basic objective of American football is for each team to gain points by moving the ball into enemy territory and either scoring a touchdown or field goal. The offensive team can advance the ball downfield by either running or passing it, while the defensive team tries to limit the forward movement of the offense by tackling players and blocking passes. If the offensive team fails to move the ball 10 yards in four plays, the ball is caught mid-play by the defense, or the offense drops the ball and it is recovered by the defense, then the offense loses possession of the ball (turnover), and the two teams swap offensive/defensive roles.

The National Football League (NFL) is one of the most popular and profitable sports leagues in the world. The 32 teams in the NFL play 17 regular season games. Analyzing NFL team statistics can make game planning more efficient and useful for the competing teams. As a very popular source of sports entertainment, the NFL also attracts millions of betters, and by analyzing key features of a team's gameplay, one can generate results that can increase the chances of making successful bets on NFL games.

Hypotheses

We hypothesized that we could train supervised machine learning models using K Nearest Neighbors, Linear Regression with Gradient Descent, Regression Trees, and Random Forests to accurately predict the number of regular season wins an NFL team will have given certain offensive and defensive team statistics.

Learning Methods

K Nearest Neighbors: K Nearest Neighbors (KNN) is a simple supervised machine learning algorithm that can be used to solve both classification and regression problems. The idea of KNN is to group closer values to one another. Our implementation measures proximity and assigns weights accordingly using Euclidean distance. We chose to implement KNN because it is relatively simple, versatile, and offers a good guideline to compare models created from other supervised learning methods.

Linear Regression with Gradient Descent: Linear Regression estimates a linear relationship of the data by equation $y = \beta_0 + \beta_1 x$. Estimating a linear trend between a team's wins and offensive and defensive performance has error, or loss, calculated using Mean Squared Error (MSE). By applying the Gradient Descent optimization algorithm with Linear Regression, we can minimize the Loss Function and create a better linear model for our data. We chose to implement Linear Regression with Gradient Descent to see how well our data is represented by a linear model to our data and minimize error.

Regression Trees: Regression Trees are a type of decision tree that predict real values. Using training data, a decision tree is built by testing which attributes and the values of those attributes most accurately split the observations and predict their target values. This is done by iterating through every attribute and testing different threshold values for those attributes. At each threshold, the average target value is found for all observations with attribute values less than the threshold and for observations with attribute values greater than the threshold. Then the sum of squared errors is taken, and the attribute/threshold combination that results in the smallest sum of squared errors is chosen as the root node of the tree. This process continues for each node until either there is some minimum number of observations in a node or the sum of squared errors is not decreasing by some delta, at which the point the node becomes a leaf and the averages found are used as predictions. We are using a regression tree to see if it better fits the data than a linear model.

Random Forest: Random Forests are created from many regression trees, along with the incorporation of randomness to prevent the model from over-fitting, or being too specific to the training data. We build a "bootstrapped" dataset, which is a dataset containing random samples, with replacement, from the original training dataset. As opposed to the regression tree method where we consider all attributes when figuring out how to split a node, we randomly select a specified number of attributes for consideration. Each time we build a tree, we use a new bootstrapped dataset and consider a random subset of variables at each step, allowing for a more diversified model compared to a single regression tree. Once we finish creating the forest, we average the predictions over all the trees to get our final prediction. We are using a random forest to see if it fits the data better than a single regression tree.

Experiments

Before conducting any learning experiments, we needed to gather, parse, and process the data collected by Pro Football Reference.

The offensive statistics we analyzed were (1) pass completion, (2) adjusted net yards per pass attempt, (3) yards per rush attempt, (4) % of drives ending in an offensive turnover, (5) % of drives ending in an offensive score, and (6) average points per drive

The defensive statistics we analyzed were (1) Quarterback (QB) pressures per drop back, (2) % of pass attempts intercepted, (3) % of opponent offensive drives ending in a turnover, and (4) % of opponent offensive drives ending in a score.

Since we are predicting a real number (the number of NFL teams' regular season wins), we use regression supervised learning methods. For each method, we used 2021 season data for training and 2020 season data for testing.

In Experiment 1, we implemented K Nearest Neighbors using weighted distances and setting K=5.

In Experiment 2, we implemented Linear Regression using Gradient Descent with a learning rate of $\alpha = 0.00005$ and a stopping condition of $\Delta w = 0.00005$.

In Experiment 3, we implemented Regression Trees with a minimum number of observations per leaf node of 2 and a stopping condition of $\Delta ssr = 0.00005$.

In Experiment 4, we implemented Random Forests, using an attribute subset count of 3 (the square root of the number of attributes) and a tree count of 1000.

Results

Figure 1 on Page 4 shows the predicted number of wins for each team created by each learning method.

Figure 2 on Page 4 shows the Root Mean Square Error (RMSE) generated by each learning method.

Analysis

As seen from the table, all our models performed fairly well and similarly. They all had around a RMSE of 2.8, and considering each NFL team only plays 17 games in a regular season, this translates to about a 16% error. When looking at the weights derived from the gradient descent model, by far the most impactful attribute was adjusted net yards per pass attempt. Surprisingly, the attributes with the least impact were percentage of drives ending in an offensive score and pass completion. When looking at the regression tree, it is a little less obvious as to what attributes split the data best, but QB pressures per drop back and the percentage of opponent offensive drives ending in a score were used for three separate nodes as the splitting attribute. When looking at the random forest, it is important to note that while results may vary due to utilizing randomness, the RMSE, on average, is still the lowest compared to the other models.

Literature Review & Related Work

In contrast to our experiments trying to predict the number of wins a football team can expect in a season, Bouzianis conducted a study using logistic regression to predict the win rates. He created a model for each of the 32 teams in the NFL using data from 2001-2016 for training. He then used those models to predict the winner of each NFL game of the 2017 and 2018 seasons. He found that accuracy of the models varied greatly between teams, ranging from 75% to 25% accuracy. Bouzianis hypothesized this is

because the consistency in team performance varied greatly between the seasons he used for training.

Baldi analyzes the use of gradient descent algorithms to train neural networks. He focuses on four different kinds of algorithms that can be used to compute the gradient: numerical integration of the system, using the explicit solution of the system, and adjoint methods in which an "auxiliary N-dimensional adjoint system" is constructed, and backpropagation. He also discusses some of the limitations of gradient descent algorithms, stating they can fail in numerous methods such as by reaching poor local minima or getting stuck in long plateaus. Baldi writes that the local minima problem can potentially be solved in larger networks as the number of local minima tend to decrease as the number of units in the network increase.

Sipper and Moore explore the idea of conservation machine learning using random forests as a case study. They define conservation machine learning as "[conserving] models across runs, users, and experiments". They state that millions of machine learning models are created, and the vast majority are discarded, and they argue that these models should instead be conserved and shared with the machine learning and data science community. They use random forests as an example to show how using many thousands of models that have already been created greatly improve accuracy and performance. They state that the extra time needed to compute the output of the many forests is minimal when compared to how long it would take to create the forests from scratch.

Zhang discusses the fundamentals of the KNN machine learning method, its limitations, and evaluation the outcomes. In the evaluation of the performance of his model, he keeps track of true positive, true negatives, false positives, and false negatives, analyzes the kappa statistic, and utilizes the average accuracy equation and measures of sensitivity (measuring correctly identified positives) and specificity (measuring correctly identified negatives). In his discussion, Zhang also studies how to optimize the value of parameter k and finding the balance between overfitting and underfitting. Zhang's work offers good information for any future studies considering usage of the KNN method.

Another extremely popular sport worldwide is football (soccer), in which Almulla and Alam analyzed individual players' performance in football matches of the Qatar Stars League (QSL) to reveal key performance metrics that contribute to wins. In researching the identification of the best performance metrics that may lead to winning a football match in QSL, they heavily focused on data collection and feature engineering by grouping certain metrics together based on their data analyses and using feature normalization with min-max normalization. Almulla and Alam use KNN, Decision Trees, Random Forest, and Linear Regression, as well as many other supervised learning methods. Their proposed model was able to reach up to 80% accuracy in identifying the winning team from the losing team.

Ahmed, Rizaner, and Ulusoy explore tree pruning with Bayes minimum risk to combat the common over-fitting problem with decision trees. Their proposed method uses a bottom-up approach by converting a parent node of a subtree to a leaf node if the estimated risk-rate of the parent node for that subtree is less than the risk-rates of its leaf. Risk-rates are estimated using the Bayes minimum risk. The results of their experiments show that pruning using their proposed method produces better classification accuracy and has satisfactory performance in terms of precision score, recall score, TP/FP rate, and area under ROC, compared to both Reduce-Error Pruning (REP) and Minimum Error Pruning (MEP), while not creating additional complexity than REP and MEP.

Conclusion & Future Work

Using various supervised learning methods, we were able to successfully create models that could predict regular season wins using team offense and defense statistics, with Random Forest having the lowest RMSE and KNN having the greatest RMSE.

However, there is still room for improvement. We could increase accuracy by using in-depth feature engineering and optimizing our machine learning algorithms by find the best hyperparameters for each method. We could improve our work on regression trees and random forests by creating additional forests, conducting forest evaluation, using "bagging" (bootstrapping and using the aggregate to create a dataset), optimizing the number of random attributes to consider for splits, and utilizing tree pruning.

Figures

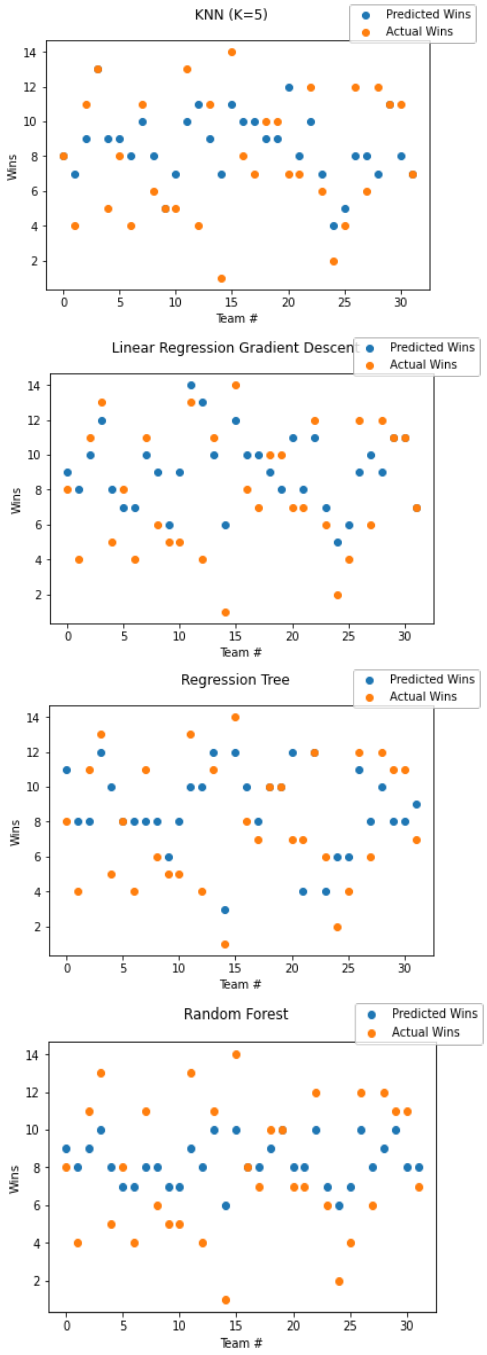

Figure 1: Predicted number of wins for every team, generated by each supervised learning method

KNN	Gradient Descent LR	Regression Tree	Random Forest
2.89	2.86	2.78	2.54

Figure 2: Root Mean Square Error (RMSE) generated by each supervised learning method.

Contributions

Arjun primarily worked on the KNN, Linear Regression with Gradient Descent, and Regression Trees learning methods, and wrote the first three literature reviews.

Daniel primarily worked on the Random Forest and Regression Trees learning methods, and wrote the last three literature reviews.

Code

https://github.com/JuneyBoy/Football-SL-Project

References

Ahmed A. M., Rizaner A., Ulusoy A. H. (2018).W A novel decision tree classification based on post-pruning with Bayes minimum risk. *PLOS ONE*, 13(4): e0194168. https://doi.org/10.1371/journal.pone.0194168

Almulla, J., & Alam, T. (2020). Machine Learning Models Reveal Key Performance Metrics of Football Players to Win Matches in Qatar Stars League. *IEEE Access*, 213695–213705. https://doi.org/10.1109/ACCESS.2020.3038601

Baldi, P. (1995). Gradient descent learning algorithm overview: a general dynamical systems perspective. IEEE Transactions on Neural Networks, 6(1), 182–195. https://doi.org/10.1109/72.363438

Bouzianis, S. (2019). Predicting the Outcome of NFL Games Using Logistic Regression. *Honors Theses and Capstones.* 474.

Sipper, M., Moore, J.H. (2021). Conservation machine learning: a case study of random forests. *Sci Rep 11*, 3629. https://doi.org/10.1038/s41598-021-83247-4

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11). https://atm.amegroups.com/article/view/10170