Predicting NFL Wins Using Interpretable Machine Learning

Joel Liebert
University of Tennessee at Chattanooga
CPSC 4430: Introduction to Machine Learning
Chattanooga, TN
gfy217@mocs.utc.edu

***Abstract*—American football games in the National Football League (NFL) have many moving parts and produce mass amounts of data. Due to the widespread interest in the sport, this data is well-kept, easily accessible, and can be used for analysis and forecasting. This analysis uses five classification algorithms to predict the winner of NFL games and seeks to answer the following questions: which classification algorithm is best suited to predict winners of NFL games, and are we able to use interpretable machine learning to understand why?**

***Keywords*—Machine Learning, Classification, Interpretable Machine Learning, National Football League, American Football**

I. Introduction

The NFL consists of 32 teams that play 16-game seasons (except for 2021, which was the first 17-game season). This equates to 512 games a season not including playoffs, and because each team fields 22 starters, there is ample opportunity for mass amounts of data to be collected. The open-source R package nflfastR provides a convenient set of functions for accessing NFL play-by-play data all the way back to 1999. This data can be used for in-depth analysis on players, teams, and seasons, and is essential for forecasting future results as well. This paper feeds this data into five binary classification algorithms to predict the winners of games. Once the models have been trained, they are ranked against each other to determine the most accurate model. Finally, using the iml (interpretable machine learning) package, the 'hood' of an algorithm is lifted to expose what is happening under the surface.

II. Motivation

A. NFL Betting Market

The sports betting market is huge and growing every year. According to the American Gaming Association, an estimated 45 million American were expected to bet on NFL games in 2021, spending over $12 billion on this gambling venture. This makes any model that predicts winners of great interest to 18% of the nation's population, especially if the model can outperform the market. Before each game, Vegas sets betting odds that effectively predict the winner; the models developed in this analysis seek to accurately predict the winner at a higher rate than Vegas, thus becoming invaluable in the sports betting market.

B. Model Selection

In early April 2022, I spoke to a data scientist mentor of mine. Among the questions I asked him was how data scientists should go about selecting algorithms and whether that comes with experience. According to him, the best way to select algorithms is to simply test as many as one can for each situation and choose the best performing model. This paper implements that approach; instead of choosing an algorithm in advance, five different ones were implemented and compared to each other to determine the best performing one. The hope is that making this a practice will eventually result in an intuition of which models are best suited for each situation.

C. Interpretable Machine Learning

In the process of interviewing for a data science position, an interviewer asked me my process for balancing model performance and model interpretability. This underscores a key point of tension within the data science world; the more complex and opaque models become, the greater their accuracy increases, in general. However, with this increase in accuracy comes a decrease in the ability of data scientists to understand the model and convey that understanding to decision makers, which can undermine trust in the model and makes data science less effective. This makes model interpretability a key issue for data scientists. There are a few packages that seek to address this issue; in R, the most robust model interpretability package is iml, which stands for 'interpretable machine learning.' This paper uses said package to help understand how the model weights each feature and makes its predictions. A thorough utilization of interpretable machine learning can help ease the tension between model performance and interpretability as it can uncover the inner workings of opaque models and help data scientists understand how they are working.

III. Methodology

A. Data Preprocessing

All data was pulled using the open-source R package 'nflfastR' which is available on CRAN. Because the data is recorded at the single play level, a significant amount of aggregation was necessary. The first step was to aggregate it by team, by game, and by season, so that the game-by-game results for each team could be easily accessed, thus mimicking box scores. After this, some clever manipulation was necessary to obtain the season stats prior to each game. For example, the week one matchup between the Arizona Cardinals and the Tennessee Titans contained zero season data, as there were no prior games available for the teams to record data. Because of this, all week one matchups were excluded from the matchup. In contrast, Arizona's week two matchup with Minnesota contained one week of season data for each team, their week three matchup with Jacksonville contained two weeks of season data, etc. An expectation of this study is that the model would predict later weeks more accurately than early weeks as later weeks have more season data stored.

B. Data Scaling

Two methods of data scaling, standardization and normalization, were employed on the data in addition to keeping the features in their raw form. In standardization, the data is transformed in terms of standard deviation from the mean, i.e., a value two standard deviations higher than the mean for that feature would be represented as 2. Normalization, on the other hand, represents values on a scale of 0 to 1, where 0 is the smallest value in that feature and 1 is the largest. These methods of feature scaling are important to avoid naturally larger features dominating smaller features (e.g., yards gained vs fumbles). I chose to implement all three of these methods to gain a better intuition of how they change model performance as well as to see which one results in the best model.

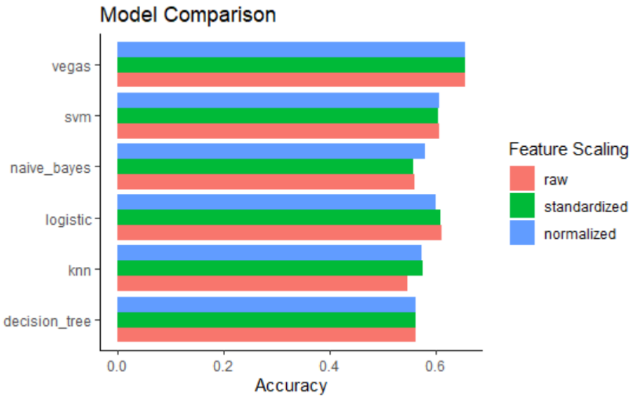
C. Model Selection

After some research, it became evident that there are five main binary classification algorithms relevant to this analysis: logistic regression, k-nearest neighbors, support vector machine, decision tree, and naïve bayes. Each algorithm has its own strengths and weaknesses; for example, decision trees are usually easy to understand and naïve bayes tends to perform poorly when the features have high multicollinearity. Instead of taking extra time to study each algorithm, examine the features, and determine which algorithm is best suited for this model, the strategy was to run all five algorithms and choose the highest performing one. This way, an error in the model selection method can be avoided and the best model will be chosen.

IV. Results

A. Model Results

The results of the five algorithms are shown below in comparison to Vegas's prediction accuracy:

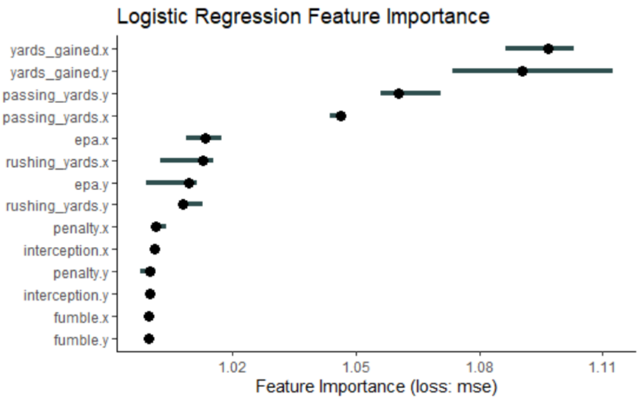


The two strongest performers are the logistic regression and support vector machine models, with the logistic model having the best overall performance using raw data, scoring an accuracy of 61%. Unfortunately, this accuracy fails to threaten Vegas's predictions which sit at 65.5% over the past 23 years.

Of interest is the lack of effect that scaling has on the models' performance. For some models, such as knn, it notably improves the performance. However, by and large it has no difference. Logistic regression performs best without any feature scaling as it prefers the data in its raw form.

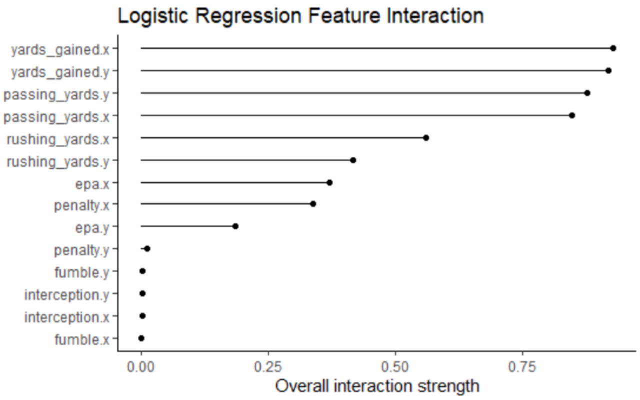
Because the logistic regression model, while using raw features, performs the best, this is the model we will examine using the iml package.

B. Interpretable ML with iml: Feature Importance

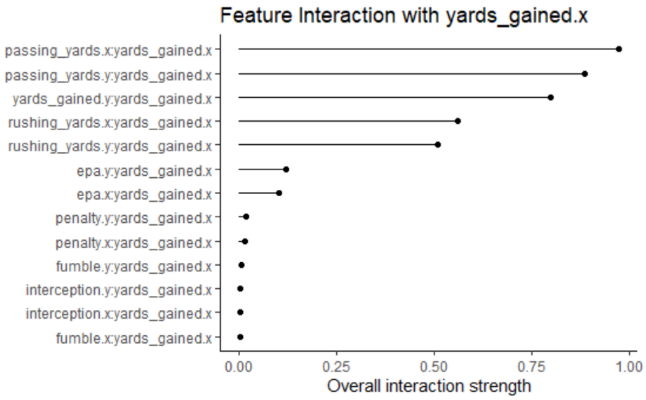


The first plot examines feature importance. This chart shows how much the model's error increases if any single feature is removed from data. By far, the two most important predictors are the home team's and away team's total season yardages (yards_gained.x and yards_gained.y, respectively). After that, passing yards and the cumulative expected points added by the home team are also important, but importance falls off significantly after that. Towards the bottom, it appears that a few features, such as fumbles, interceptions, and penalties, are not important at all. This suggest that these features are likely much more random than anything else and are not very useful for predicting winners. This chart helps the modeler understand what features are key to the model's predictions and thus helps aid its interpretability.

C. Interpretable ML with iml: Feature Interaction

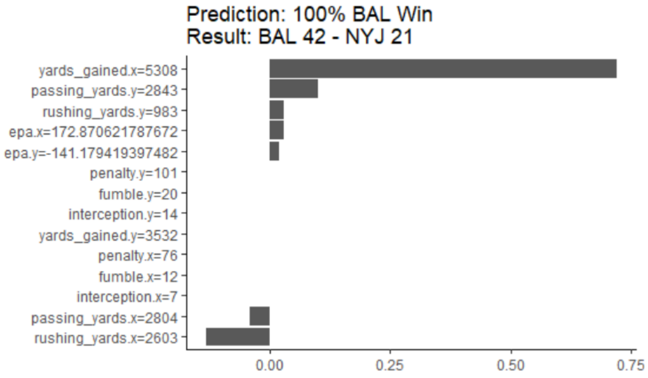


This second plot takes a closer look at how strongly each feature interacts with other features on a scale of 0 to 1, where 0 implies that feature doesn't interact with other features at all, while a 1 indicates that feature interacts with another feature completely. Here, we see a distribution remarkably similar to the feature importance chart. The yards gained for both home and away teams score extremely high feature interaction. The iml package allows us to drill deeper and see how a single feature interacts with the others. Let us examine 'yards_gained.x' more closely.



Unsurprisingly, the home team's yards gained interacts almost fully with the home team's passing yards. This might indicate multicollinearity between the two that could be addressed in the data preprocessing stage. Classification algorithms tend to perform the best with the lowest amount of maintenance when multicollinearity is at a minimum, so these two charts can help the data scientist examine his or her data and understand any interactions occurring between the features.

D. Interpretable ML with iml: Shapley Values



Shapley values are probably the coolest and most insightful look into how a model works. In essence, the Shapley values for each feature show how much that feature contributes to an individual prediction. In this case, we are examining 2019's week 15 matchup between Baltimore and the New York Jets. Based on the 14 weeks of game data leading up to this matchup, the logistic regression model predicts a 100% chance of victory for the Ravens. This prediction proves prescient as Baltimore wiped the Jets in a 21-point victory. The biggest factor in the prediction? Through 14 weeks, Baltimore accumulated 5,308 total yards, as opposed to New York's 3,512. This imbalance signaled to the model that Baltimore had an extremely strong, even inevitable chance of victory. This behind-the-scenes peek into the model that iml allows is a fascinating examination that was previously hidden from data scientists, which shows the value of interpretable machine learning.

V. Conclusion

Overall, the logistic regression model performs best. This may be due to the simplicity of the data and the ease of which logistic regression takes in a limited number of features to make its predictions. Surprisingly, it performs most accurately when the features are unscaled – this illustrates how important it is to test multiple models in multiple ways, as a more accurate way of classification may not come from the expected source. Unfortunately, it is no match for Vegas's predictions as it falls four points shy, but overall it still performs relatively well. Finally, the iml package allowed for a deeper look into how the model works. This could help improve its future performance as well as give the data scientist a better understanding of it, which allows him or her to communicate the findings more effectively.

References

[1] *Record 45.2 million Americans to Wager on 2021 NFL season*. American Gaming Association. (2021, September 7). Retrieved May 1, 2022, from https://www.americangaming.org/new/record-45-2-million-americans-to-wager-on-2021-nfl-season/

[2] Comprehensive R Archive Network (CRAN). (2021, October 6). *Functions to efficiently access NFL play by play data [R package nflfastr version 4.3.0]*. The Comprehensive R Archive Network. Retrieved May 1, 2022, from https://cran.rstudio.com/web/packages/nflfastR/index.html