The Application of Machine Learning to Predict NFL Running Back Performance
STATS-5405

Giovanni Lunetta[a,*], Sam Lutzel[a,*]

[a] *University of Connecticut, Statistics, 2075 Hillside Road, Storrs, 6269*

Abstract

This study employs an ensemble of machine learning techniques, including Multiple Linear Regression (MLR), Generalized Linear Models (GLM), and XGBoost to enhance predictive analytics in the National Football League (NFL). The research focuses on one of the most critical aspects of football offense - the running game. In order to predict the yards gained by NFL running backs following a handoff, the study analyzes a plethora of player tracking data from the 2017 to 2019 seasons. It integrates a variety of factors such as player positions, orientation, and the situational context of the game, including down, distance, and field position. This multifaceted approach aims to yield highly accurate models that can serve as valuable tools for coaching staffs to optimize play-calling, manage player workloads, and enhance game planning. The predictive insights derived from this research are intended to support teams in deploying their running backs more effectively, leading to potentially improved outcomes on the field. As the NFL continues to evolve with a greater emphasis on analytics, this study seeks to contribute significantly to the field of sports analytics by providing a model that underscores the importance of the running game in a predominantly pass-oriented league.

Keywords: Machine Learning, Predictive Analytics, Sports Analytics

1. Introduction

1.1. Background of the National Football League

The National Football League (NFL) is the most popular professional sports league in the United States, with an estimated 187 million fans. The league consists of 32 teams divided into two conferences, the American Football Conference (AFC) and the National Football Conference (NFC). Each conference is further divided into four divisions, with four teams in each division. The NFL season consists of 17 weeks, with each team playing 16 games and having one bye week. The regular season is followed by a 12-team playoff, with the winner of each conference advancing to the Super Bowl, the league's championship game.

The NFL game strategy primarily consists of two ways to advance the football down the field: running and passing. Running the ball is defined as a play in which the quarterback gives the ball to another player that is located behind them (a handoff or a small toss). The player then attempts to run the ball down the field as far as possible before being tackled to the ground. Passing the ball is defined as a play in which the quarterback throws the ball to another player down the field. The player then attempts to catch the ball and advance it down the field as far as possible before being tackled to the ground. The goal of each team is to score as many points as possible by advancing the ball down the field and into the end zone or by kicking the ball through the field goal posts. The team with the most points at the end of the game wins.

*Corresponding author

Email addresses: giovanni.lunetta@uconn.edu (Giovanni Lunetta), samuel.lutzel@uconn.edu (Sam Lutzel)

1.2. Motivation

The primary goal of this research is to develop cutting-edge predictive models capable of accurately estimating the yards a running back will gain after a handoff during NFL games. This objective is pivotal for formulating advanced offensive strategies and refining player evaluations. The running play is a fundamental aspect of the game that can dictate the tempo, control the clock, and establish physical dominance.

The motivation behind this study stems from the transformative impact that data analytics has had on sports, particularly in the NFL, where the fusion of technology and sports science has begun to redefine how the game is played and understood. In an era where marginal gains are increasingly sought after, the ability to predict the outcome of running plays with high precision can provide teams with a significant competitive advantage. It enables coaches to make informed decisions regarding play selection, player rotations, and game management, especially in critical moments of a match. Additionally, the insights from this study can empower front offices in their scouting and drafting processes by quantifying the expected value a running back adds to their team. Furthermore, there is also a tremendous opportunity to leverage these predictive models on the defensive end of the ball, allowing teams to better anticipate and defend against running plays. With better insights into the opponents running game, defenses can adjust their schemes and personnel to counter the opposing team's offensive strategy, such as by stacking the box or blitzing the quarterback. In addition to the benefits performance analytics provides to the teams, it also helps fans select better fantasy football teams and make more informed betting decisions. This leads to a more engaging and enjoyable experience for the fans, which is critical for the long-term success of the league.

Ultimately, the true beauty of this research lies in its ability to bridge the gap between complex player tracking data and practical on-field strategies. In specific, it's about enhancing the very essence of the game and enriching the broader discourse on sports performance analytics. By pioneering research in this domain, this study is set to propel the analytical capabilities of NFL teams to new heights, providing them with tools that were once considered unimaginable. Ultimately, it contributes to the ongoing evolution of the sport itself, marking a pivotal moment in the history of football and the broader world of sports analytics.

2. Methods

2.1. Data Cleaning and Preprocessing

Data cleaning and preprocessing are critical to ensuring the quality and integrity of machine learning models. The cleaned dataset was obtained by addressing any inconsistencies and handling missing data through imputation techniques. Preprocessing steps included encoding categorical variables using one-hot encoding or label encoding methods and scaling and normalizing numerical variables to ensure they are on comparable scales.

To begin, several variables were removed in order to simulate the beginning of each play. For instance, the dataset include the speed, acceleration, direction, and distance traveled of each player on the field at the beginning of each play. However, this information is not available to the coaching staff prior to the snap of the ball. Therefore, these variables were removed from the datset. Furthermore, some variables may introduce multicollinearity issues since they represent the same information. For instance, the dataset includes the name and id of each player on the field. However, these variables are highly correlated with each other. Therefore, only one of these variables was kept in the dataset.

When the dataset was first obtained, each row represented a single player on the field. However, the goal of this study is to predict the yards gained after a handoff for each play. Therefore, every 22 rows had to be merged into one row (11 players on each side). During this process, any duplicated information/columns were dropped from the dataset. One example of this is the weather during the play. When all 22 rows are merged, it included the weather 22 times, even though weather is only needed once.

In addition to the attempt at removing unnecessary variables and proactively removing potenital multicollienarity issues, variables such as weather, wind speed, and wind direction had multiple levels, some of which represented the same data. For instance, “rainy” and “showers” were considered the same weather type.

There were multiple instances of relationships between inputs on the same variable that existed similar to the example above. Due to this, these weather types were grouped into one category - “rain.” This helps to drastically reduce the complexity of the dataset. These variables were also converted to factors for the modeling process.

Variables such as offense personnel had to be converted to dummy variables. The input for each dummy variable was either 0 or 1 depending on whether or not the offense had that personnel on the field. For instance, if the offense had 2 running backs on the field, the input for the “RB” dummy variable would be 2. The input for the “WR” dummy variable would be 0 since there were no wide receivers on the field. This process was repeated for all personnel types.

One additional step that was taken was the randomization of all plays within the dataset. The reason for this randomization is to ensure that each row (play of a game) is independent of one another. Furthermore, the dataset only contains running plays. Therefore, passing plays that occured between the running plays were removed. In other words, all plays are not dependent on the previous outcome. In all, the randomization of the dataset in conjunction with the removal of passing plays ensures that each row is independent of one another.

2.2. Data Exploration

Prior to the development of the predictive models, a comprehensive exploratory data analysis (EDA) was conducted to understand the underlying structure and characteristics of the data. This step involved visualizing the distribution of key variables, identifying patterns and outliers, and exploring correlations and interactions between predictors. Data visualization, through techniques like scatter plots, histograms, and heatmaps, can offer an intuitive understanding of data distributions, correlations, and potential clusters within the dataset. EDA-informed feature selection and engineering strategies highlighted potential predictors that are most informative of yards gained after a handoff.

2.3. Feature Engineering

During preprocessing, we will also undertake feature engineering to create new variables that may have a stronger relationship with the target variable. This could include interaction terms that capture the combined effect of two predictors, polynomial features for capturing non-linear relationships, and domain-specific features that encapsulate strategic elements of the game.

2.4. Model Development and Evaluation

With a clean and prepared dataset, we will then proceed to develop our MLR, GLM, and XGBoost models.

2.5. Multiple Linear Regression (MLR)

Multiple linear regression models predict a continuous repsonse variable using a linear combination of predictors. For our baseline MLR model, we will use the ordinary least squares (OLS) method to estimate the coefficients of our predictor variables. The model is specified as:

$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_j X_{ij} + \epsilon_i$

where Y_i represents the yards gained after the handoff for the i^{th} observation, X_{ij} is the i^{th} observation on the j^{th} predictor variable (where $j = 1, ..., p$), β_j is the j^{th} coefficient to be estimated corresponding to the proper X_{ij} variable, and ϵ_i is the error term. The j^{th} coefficient, β_j, represents the change in the response variable for a unit change in the X_{ij} predictor variable, while holding all other predictors constant.

The ordinary least squares (OLS) method will be deployed to minimize the sum of the squared differences between the observed and predicted values. The optimization problem can be represented as minimizing the sum of errors squared:

$$\Sigma_{i=0}^{n} \epsilon_i^2 = \Sigma_{i=0}^{n} (Y_i - \hat{Y}_i)^2 = \Sigma_{i=0}^{n} (Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_j X_{ij})^2$$

Here, $\Sigma_{i=0}^{n}\epsilon_i^2$ represents the sum of the squared residuals, which we aim to minimize. The variable $\hat{Y}_i$ represents the predicted outcome of the model for the i^{th} observation. This approach assumes that the relationship between the independent variables and the dependent variable is linear. To ensure the robustness of our model, we will conduct a series of diagnostic tests:

1. Linearity: We will use scatter plots and residual plots to verify that the relationship between the predictors and the response is linear.
2. Homoscedasticity: We will inspect the residuals to confirm constant variance across all levels of the independent variables. This can be assessed visually using a residual vs. fitted values plot.
3. Independence: The Durbin-Watson test will help in detecting the presence of autocorrelation in the residuals, which should not be present in the data.
4. Normality of Residuals: Normality will be checked using Q-Q plots and statistical tests like the Shapiro-Wilk test.

If any assumptions are violated, we may consider transformations of variables or the use of robust regression techniques.

2.6. Generalized Linear Models (GLM)

GLMs extend the linear model framework to allow for response variables that have error distribution models other than a normal distribution. They are particularly useful when dealing with non-normal response variables, such as count data or binary outcomes. In its general form, a GLM consists of three elements:

1. Random Component: Specifies the probability distribution of the response variable Y, such as normal, binomial, Poisson, or exponential.
2. Systematic Component: A linear predictor $\eta = X\beta$.
3. Link Function: A function g that relates the mean of the response variable $E(Y)$ to the linear predictor η.

The choice of link function is crucial and is typically selected based on the nature of the distribution of the response variable. For instance, a logit link function is used for a binomial distribution, and a log link function is often used for a Poisson distribution.

The likelihood function for a GLM can be written as:

$$L(\beta) = \prod_{i=1}^{n} f(y_i; \theta_i, \phi)$$

where $f(y_i; \theta_i, \phi)$ is the probability function for the i-th observation, θ_i is the parameter of interest (e.g., mean), and ϕ is the dispersion parameter. The goal is to find the values of β that maximize this likelihood function.

2.7. XGBoost

XGBoost stands for eXtreme Gradient Boosting and is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. It is a powerful technique that can handle a variety of regression and classification problems. For regression, it can be configured to optimize for different loss functions; the most common for regression being the squared error loss:

$$L(\theta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i are the observed values, and $\hat{y}_i$ are the predicted values.

In XGBoost, each new tree is built to correct the errors made by the previous ones. The algorithm combines weak predictive models to form a strong predictor. The model's complexity is controlled by the regularization term $\Omega(\theta)$ which is a function of the tree structure and the number of leaves. The overall objective function to be minimized is:

$$\text{Obj}(\theta) = L(\theta) + \lambda \sum_k (w_k^2) + \gamma T$$

where w_k represents the leaf weights of the trees, T is the number of leaves, λ is the L2 regularization term on the weights, and γ is the complexity control on the number of leaves. For regression tasks, we can also utilize the quantile loss which is particularly useful for prediction intervals:

$$L_\tau(\theta) = \sum_{i=1}^{n} \left[\tau(y_i - \hat{y}_i)\mathbb{1}_{y_i \geq \hat{y}_i} + (1-\tau)(\hat{y}_i - y_i)\mathbb{1}_{y_i < \hat{y}_i} \right]$$

Here, 1 is an indicator function, and τ is the quantile of interest, allowing us to model different parts of the conditional distribution of the response variable.

XGBoost also provides a feature importance score, which is a metric that quantifies the contribution of each feature to the model's predictive power. This is done by measuring the impact on the model's accuracy each time a feature is used to split the data across all trees.

2.8. Model Performance Metrics

For MLR and GLM, the model's performance will be evaluated using metrics such as the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). For the XGBoost model, along with MSE and MAE, we will assess performance using additional metrics like the R-squared for regression tasks and feature importance scores to understand which variables are most predictive.

2.9. Model Interpretation and Application

The final step will be to interpret the models in the context of NFL games. This will involve translating the statistical outputs into actionable insights for coaches and team strategists, providing recommendations on how to leverage the results for competitive advantage in play-calling and player evaluation.

2.10. Software and Tools

All analyses will be conducted in R, a statistical computing language that provides a wide array of packages for machine learning and data analysis. For MLR and GLM, we will utilize the `stats` package that comes with the R base installation. For our XGBoost model, we will use the `xgboost` package, which is specifically designed for speed and performance. Data manipulation and cleansing will be managed with packages like `dplyr` and `tidyr`, while `ggplot2` will be employed for data visualization to facilitate understanding and interpretation of the data and model outputs. For feature engineering and preprocessing, we will take advantage of `caret` or `recipes`. Hyperparameter tuning can be optimized using the `tune` package, and for cross-validation, the `rsample` package will be employed. The `broom` and `modelr` packages will be useful for tidying model outputs and working with models in a pipeline, respectively. This suite of packages will enable a comprehensive workflow within R for developing, evaluating, and interpreting the predictive models.

3. Equations

Here is an equation:

$$f_X(x) = \left(\frac{\alpha}{\beta}\right)\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}}; \alpha, \beta, x > 0.$$

Inline equations work as well: $\sum_{i=2}^{\infty}\{\alpha_i^\beta\}$

4. Figures and tables

Figure 1 is generated using an R chunk.

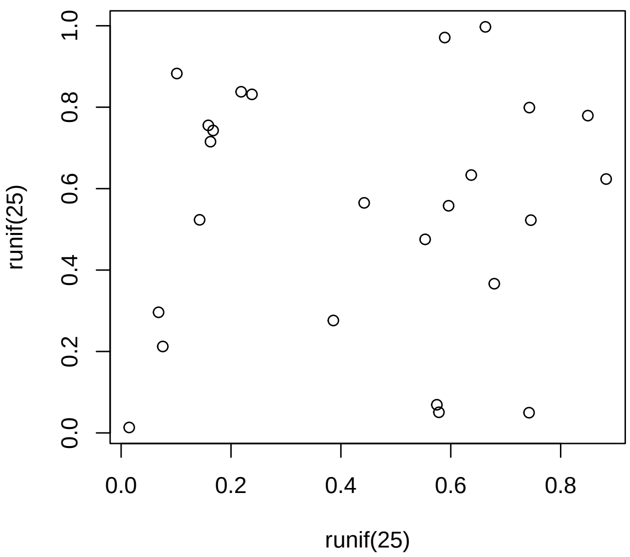

Figure 1: A meaningless scatterplot

5. Tables coming from R

Tables can also be generated using R chunks, as shown in Table 1 example.

```
knitr::kable(head(mtcars)[,1:4])
```

Table 1: Caption centered above table

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
Hornet Sportabout	18.7	8	360	175
Valiant	18.1	6	225	105

References

6. Bibliography styles

Here are two sample references: **(author?)** [1] **(author?)** [2].

By default, natbib will be used with the `authoryear` style, set in `classoption` variable in YAML. You can sets extra options with `natbiboptions` variable in YAML header. Example

`natbiboptions: longnamesfirst,angle,semicolon`

There are various more specific bibliography styles available at https://support.stmdocs.in/wiki/index.php?title=Model-wise_bibliographic_style_files. To use one of these, add it in the header using, for example, `biblio-style: model1-num-names`.

6.1. Using CSL

If `cite-method` is set to `citeproc` in `elsevier_article()`, then pandoc is used for citations instead of `natbib`. In this case, the `csl` option is used to format the references. By default, this template will provide an appropriate style, but alternative `csl` files are available from https://www.zotero.org/styles?q=elsevier. These can be downloaded and stored locally, or the url can be used as in the example header.

References

[1] R. P. Feynman, F. L. Vernon Jr., The theory of a general quantum system interacting with a linear dissipative system, Annals of Physics 24 (1963) 118–173. doi:10.1016/0003-4916(63)90068-X.

[2] P. A. M. Dirac, The Lorentz transformation and absolute time, Physica 19 (1-–12) (1953) 888–896. doi:10.1016/S0031-8914(53)80099-6.