NFL Play Call Prediction Using Sequential Neural Networks

Joseph W. Director
University of Colorado Denver
MS Statistics

Joshua French
University of Colorado Denver
Advisor

Steffen Borgwardt
University of Colorado Denver
Graduate Committee

Florian Pfender
University of Colorado Denver
Graduate Committee

Spring 2022

Abstract

The prevelance of data analytics in professional sports has significantly increased over the last 20 years. First popularized in *Moneyball: The Art of Winning an Unfair Game (2003)*, the use of advanced analytics is now mainstream in the four major U.S. sports and abroad. In the National Football League (NFL), millions of dollars are invested into analytics departments and data is being used to drive decision making at every level of an organization's operation. These departments can leverage statistical methods to learn the opposition's tendencies, providing a substantial competitive advantage. In particular, the defensive team can improve its strategy by accurately predicting the offensive team's play call (whether the play is a run or a pass). To this end, many prior works have implemented machine learning algorithms for play call prediction. However, none of the works encountered have treated play-by-play data as sequential. In Football, the offensive team's current play call is dependent upon the sequence of plays called before; therefore, there is a time series component that a modeling strategy must account for. In this work, we explore the ability of sequential deep learning models to predict NFL play calls. Namely, we compare the performance of Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks to baseline models (Logistic Regression and Gradient Boosted Decision Trees). Using classification accuracy and ROC-AUC as metrics, we found that sequential models outperform the baseline.

Contents

1 Introduction

1.1 Background

Gameplay of NFL football is seperated into a sequence of instances called plays. The two teams on either side of the ball are allowed to reposition themselves and prepare in betweeen these instances. Given this nature, there is immense opportunity to strategize when the game is not in play. Akin to moving pieces on a chessboard, the coaching staff decides their team's best course of action by anticipating their opponent's moves. For regular plays (i.e. not a kick-off, punt, or field goal attempt), there is a binary option for the type of play the offensive team can do; either a pass or a run. A strategic advantage is gained for the defensive team if they can accurately predict this outcome. As a simplistic example, if they predict pass they can put more players in pass coverage, or put more players near the line of scrimmage if they predict run.

There are a number of indicators that can inform play call prediction. Certain personnel packages (groups of players from various positions) and formations of the offensive team are more associated with either passes or runs. Unfortunately, the NFL does not publicly release data containing specific personnel or formations. However, there exists a general binary indicator for the formation; whether the play was from the shotgun (QB lines up a few yards back from the center) or under-center (QB lines up directly behind the center). Beyond this, there is the in-game context of the current play. This includes the down (how many plays can be used to gain the required distance), the distance (the amount of yards needed to gain in order to keep the ball), the score differential, the amount of time remaining, etc. The conditions of these factors all incentivize the offensive team to use either a run or a pass play. For example, if an offensive team is losing by a lot of points with little time remaining in the game, they are more likely to pass because they can gain more yards using less time. Lastly, tendencies of the offensive team can be studied. This is done by accumulating the relative frequency of passes to runs for the offensive team (pass to run ratio) as well as how successful they are at either passing or running (average yards gained per run or pass play).

An NFL coach combines experience and intuition to predict the play call. In this work, experience is replaced by labeled data points from the entirety of a single NFL season (2019-2020), and intuition by a supervised machine learning algorithm. A supervised machine learning task involves teaching a computer to learn the underlying patterns relating the response variable to the features. If the model can discern information about the features and response well enough, its predictions should generalize well to unseen instances. A popular domain of machine learning is the field of deep learning. Deep learning uses artificial neural networks (ANNs) that loosely resemble a biological brain. They contain networks of individual neurons or nodes, each with its own activation signal, that is each capable of sending and receiving signals to other nodes (through weights). Types of deep learning algorithms vary in complexity and structure. Here we examine recurrent neural networks (RNNs) and their variant, long short term memory (LSTM) networks. These kinds of networks were originally designed for speech and text recognition because of their ability to learn sequential patterns.

1.2 Problem Statement

Prior works have approached play call prediction with machine learning, many implementing complex classification algorithms such as ensemble models and multi-layer perceptrons. These methods only provide marginal performance increases from simplistic models such as logistic regression. This is because, as is often the case in machine learning tasks, the signal relating the response to the features is only so strong. Therefore, in these cases, the choice of a complex algorithm provides small return as there is little that the added complexity can pick up on. Overcoming these performance limits requires rethinking how the data itself is structured. In prior approaches, each singular play is treated as a sample point. Treating the data this way means the model isn't aware of any sequential patterns that may exist. In the approach proposed by this work, an individual sample point is represented by a play sequence of length k.

Table 1: Feature Descriptions

Feature	**Description**	**Type**
Posteam	Name of team on offense for the current play	Categorical
Defteam	Name of team on defense for the current play	Categorical
Yardline	Distance from the goal line (yards)	Numeric
Seconds remaining	Amount of time remaining in current half	Numeric
Yards to go	Yards needed to gain for a first down	Numeric
Down	Number of plays to get a first down	Categorical
Shotgun	Whether the offensive team lines up in shotgun formation	Binary
No huddle	Whether the offensive team used the huddle before the snap	Binary
Posteam timeouts	Timeouts remaining for offensive team	Numeric
Defteam timeouts	Timeouts remaining for defensive team	Numeric
Score differential	Difference in score between offensive and defensive team	Numeric
Temperature	On field temperature for the current play	Numeric
Windspeed	On field windspeed for the current play	Numeric
Posteam Home	Whether the offensive team is on its home field	Binary
Half	Whether the play is ran in first or second half	Binary
Dome	Whether the field has a dome or not	Binary
Open	Whether a domed field is open or closed	Binary
Outdoors	Whether the game is played outdoors	Binary
Cumulative run yards	Total yards gained per run play (at current point)	Numeric
Cumulative pass yards	Total yards gained per pass play (at current point)	Numeric
Pass to run ratio	Current proportion of passes ran to total plays	Numeric
Pass yards allowed	Total yards allowed per pass play (Defteam)	Numeric
Run yards allowed	Total yards allowed per run play (Defteam)	Numeric

1.3 Data

Abundant play-by-play data is made easily accessible by the `nflfastR` package. Data published officially by the NFL is available so no webscraping is required as a step in preprocessing. A singular season (2019) was chosen as the sample because the offensive and defensive teams are used as features; teams are subject to change from year to year. Around 32,000 individual play instances occurred over the course of this season. A list of features chosen for modeling is included in Table 1. Some of these features are given directly by `nflfastR`, others are acquired through feature engineering. The data given is two dimensional (n, m): n samples (individual plays) and m features. This dimensions are fed into baseline models. For sequential models, three dimensional data is required (n, k, m): n samples, each sample consisting of k consecutive plays, and m features for each play in the sequence. This requires additional preprocessing.

2 Methodology

2.1 Aims

Our analysis aims to implement industry standard data science practices for building and testing models. This consists of building a scalable data pipeline for preprocessing, hyperparameter tuning, and model selection. To this end, cloud computing resources were provisioned from Google Cloud Platform (GCP). This platform allowed for distributed model training on multiple graphics preprocessing units (GPUs); a critical resource given the computational demands of model tuning. The overall workflow for the project is given by Figure 1. Additional steps such as feature engineering and data visualization are included to attain a wholistic view of the data used for modeling.

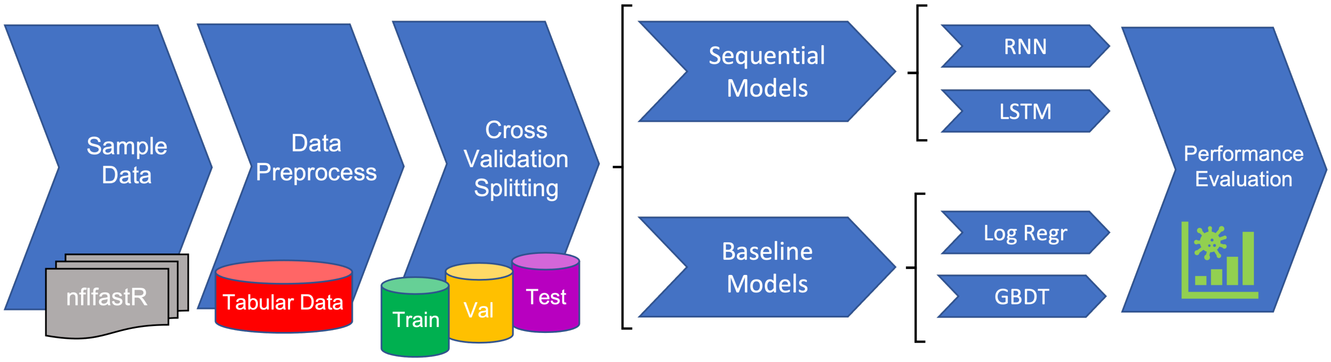

Figure 1: Overall Methodologic Workflow

2.2 Preprocessing

Data sampled directly from `nflfastR` contains all play types and outcomes from both the regular and post-season. For the purpose of this analysis, we are only interested in regular season plays that were either passes or runs. This meant removing all instances of special teams plays (kick-offs, punts, field goal attempts) and pre-snap penalties. It is important to note that some plays are intended to be passes but end up being runs (this is called a QB scramble). We consider these to count as passes since this was the intended play call.

Missing values were only encountered for some of the weather features selected. Temperature and wind are NA for games that were played inside of a dome. Inside NFL domes, the temperature is controlled and most often set to 72 degrees fahrenheit. Missing values for these features were set to 0 windspeed and 72 degrees fahrenheit for all play instances occurring inside a dome.

For all methods considered, binary and categorical features need to be one-hot encoded. This is because machine learning models aren't able to interperet text strings directly. One-hot encoding entails creating a new column for each level of a categorical variable. The column is a 1 if an instance falls under that category and 0 if not. One-hot endcoding was chosen over integer encoding to avoid creating any ordinal pattern in the categorical features.

2.3 Feature Engineering

Feature engineering is a vital aspect of the machine learning process. It is a method by which domain knowledge is leveraged to transform raw data into meaningful features that are capable of distinguishing classes in the response. In this case, features are constructed to describe historical tendencies of both the offensive and defensive teams. This information is designed to be accurate up until the current play instance.

Cumulative tendencies are not readily available in data loaded from `nflfastR`. However, enough information is included in order to build these features. For the cumulative pass to run ratio, we group the data by offensive team and calculate a running total of passes ran divided by total plays. This ensures the feature is current up to a given play; reflecting the information the defensive team would have before the snap. Similarly, we build in features communicating the offensive teams effectiveness at either passing or running. The available data includes how many yards were gained for a given play. Again, we group data by offensive team and calculate a runnning total of yards gained (for each type of play) divided by how many plays were ran of that type.

In addition to having a sense of the offensive teams tendencies and effectiveness, the defensive team will also be aware of its own weaknesses. This aspect is built into the data the same way as offensive tendencies. Instead of grouping by offensive team, we group by defensive team and add a running total of yards allowed per play type.

2.4 Cross Validation Splitting

Cross validation splitting is a method for assessing the performance of a model on unseen data. The original sample of data is partitioned into three non-overlapping subsets called training, validation, and testing. Training is used for teaching the model to learn the patterns within the sample. Validation is the first unseen partition and is reserved for evaluating which hyperparameter combinations perform best. Hyperparameters, unlike model parameters, are not learned during the training algorithm and therefore need to be pre-specified by the user. Given a number of candidate models performing well on the validation set, a final performance evaluation is done on the testing data. Metrics recorded at this stage are used for final reporting.

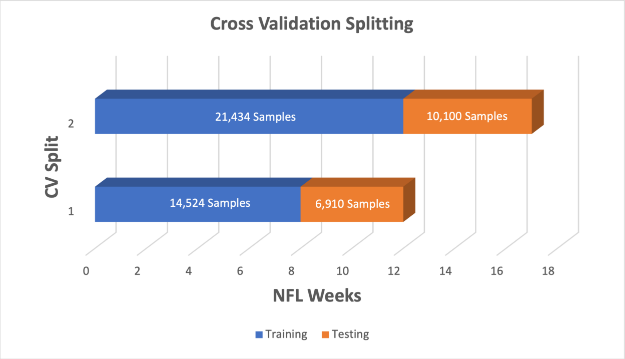

Figure 2: Cross Validation Splitting

The cross validation scheme is described in Figure 2. The first 8 weeks of the NFL season are used for the training set, with weeks 9-12 as validation for evaluating hyperparameters. For each model type, we then select the best 20 hyperparameter combinations for testing. Models with these combinations are retrained using weeks 1-12 and evaluated for final performance using weeks 13-17.

2.5 Data Visualization

Data visualization is an important step before applying any modeling strategy. At this stage, we explore the distribution of the response variable as well as its relationship to the features.

Response Variable

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.