# An Introduction to Modeling NFL Tracking Data

By Avery Ennis

Honor's Thesis, Yeshiva University

# Table of Contents

# I. Introduction

## 1A. Next Gen Stats NFL Tracking Data

In recent years, the NFL has taken steps to make an easier environment in which teams, analysts, and statisticians can perform more advanced football analysis with its Next Gen Stats initiative. In 2014 the NFL embedded tracking chips in every player's pads, allowing the league to capture each player's position, speed, and direction at many different timestamps throughout every play of every football game. In 2017, the NFL game added this tracking technology to the footballs themselves. By 2018 the NFL published this tracking data to every team, allowing teams to conduct more thorough analysis for both game preparation and roster building (NFL Operations, 2015).

In 2019, the NFL started to make parts of this data available to the public. Over the past two seasons, through an open competition hosted by the NFL called the Big Data Bowl, NFL fans have succeeded in employing advanced machine learning techniques on this tracking data to analyze the level of success of different wide receiver route combinations (NFL Operations, Big Data Bowl, 2019) and to model expected yards gained on running plays (NFL Operations, Big Data Bowl, 2020). These data advances have thrown the NFL into a data analytics revolution and many NFL team now employ statistics and machine learning experts to gain any edge possible from this newly available information.

## 1B. Current Work: EPA

Expected Points Added (EPA), first created in the 80's and significantly advanced and modernized by many in the analytics world over the past few years, has become the baseline statistic used to preform football analysis. EPA reduces each play to an individual number meant to represent how that play affects the expected point value of the offense's drive. If a play nets +1.5 EPA, that means that the play's result increased the expected value of the offense's drive by 1.5 points. The actual calculation of the drive's expected points is fairly involved and complex, but, very generally, is computed by comparing the current game situation (down, distance, quarter, etc.) to similar historical examples (Alok Pattani, ESPN Analytics, 2012).

For example, if before a play an offense's drive was worth 1.5 expected points, and after the play it was worth 1.8 expected points, the EPA of the play would be +.3. Conversely, if after the play the drive was worth 1.2 expected points, the EPA of the play would be -.3.

EPA solves a fundamental issue when analyzing football plays. Unlike other sports, in which the goals of each play are well defined and consistent, A football play's goal is uniquely intertwined with its in game situation. In baseball, for example, the batter is always attempting to score runs, and the pitcher is always attempting to prevent runs. Even if the pitching team is winning or losing by a large margin, the goals remain the same. In football, however, the success of each play is necessarily dictated by the in-game situation; A 10-yard pass on 4th and 9 is a huge success, while a 10-yard pass on 4th and 11 is a complete failure. These inconsistent goals make football analysis based on conventional statistics such as yards gained, touchdowns, and interceptions extremely difficult, as there is no guarantee that all yards and touchdowns represent the same level of success and failure. EPA solves this issue by mapping each play onto the same scale that measures success and failure given the play's context, as opposed to simply presenting the yards gained on the play. For example, EPA would successfully capture the difference in value from the two fictional 10-yard passes that illustrate the shortcomings of traditional football statistics.

As a result, EPA enables analysts to compare the effectiveness of plays, and by extension players, who definitionally play in unique contexts, making it an extremely important analytical tool for such a fundamentally context-specific sport.

## 1C. My Contribution

I have received 6 weeks' worth of Next Gen Stats tracking data from the beginning of the 2017 season (NFL Operations, 2019). Furthermore, using Maksim Horowitz's and Ron Yurko's nflscrapR package, I have obtained the EPA data for each play during those six weeks (Horowitz and Yurko, 2018). I attempted to use the tracking data to create a machine learning model that can predict the yards that will be gained from passing plays, and by extension the EPA of passing plays. (Note that my goal is fairly similar to that of the 2020 Big Data Bowl. While the Big Data Bowl was aimed at predicting the outcome of running plays, however, my focus was attempting to predict the outcome of passing plays.) Although ultimately my modeling was not as successful as I had originally hoped, I believe that is due to a relative lack of data. With more data, I believe a model that can relatively successfully predict passing plays' outcomes is attainable.

Although my efforts ultimately did not result in an overly accurate model, I will outline my process, work, and thoughts on use cases for such a model. I believe my work provides an outline for how to approach modeling NFL tracking data, and I believe a more successful model would have numerous important use cases for improving the world of football analytics.

# II. Research Methodology

## 2A. An Explanation of Machine Learning & Modeling

Before explaining my methodology, I will give a brief explanation of the general structure of machine learning models. Important technical words that will appear numerous times throughout the rest of this paper will be bolded for emphasis and clarity.

Formally, machine learning is the science of getting computers to solve a task without explicitly programming them to do so. Commonly, and in this project, that “task” is a prediction, and the goal of a machine learning model is to train the computer to accurately predict something, given other, related information.

To illustrate the function of such a prediction model, I will create a fictional dataset and walk through the process of creating such a model for that dataset. Consider a house sales dataset that, for each house sold over a period of time, contains the house’s number of bedrooms, number of bathrooms, square footage, and eventual selling price. Pretend we are tasked with using that data to predict the selling price of future house sales. In other words, our task is to find the relationship, if it exists, between the information we have about a house (the number of bedrooms, number of bathrooms, and square footage), and the house’s eventual selling price, and then to use this knowledge to predict house prices in future sales. In modeling terms, the three pieces of information we have about each house are referred to as **features**, and the price of the house is the **target variable.** The process of attempting to have the computer learn the relationship between the features and the target variable is referred to as **training.** Thus, more generally, the goal of any prediction model is to, as accurately as possible, train the model to find the relationship between a set of features and a target variable**.**

Not only, however, does a model require data for training, but it requires data for **testing.** Thus, at the beginning of the modeling process, the data is **split** into two separate data sets: a **training dataset**, and a **testing dataset.** When training a model to learn the relationship between the features and the target variable**,** it is only given access to the training dataset. Once a model is sufficiently trained, it is then tested using the testing dataset. To do so, the target variable is removed from the testing dataset, and the model is tasked with using the testing dataset’s features to predict the removed target values. The model's predictions are then compared to the actual target values, and the model's **accuracy** is determined.

As further illustration, suppose the house sales dataset has 100,000 rows, each row containing the features (number of bedrooms, number of bathrooms, and square footage) and target variable (sale price) for each of 100,000 sold houses. First, the data is split into training and testing datasets. Generally, suggested splits put around 80 percent of the data in the training dataset, so this split creates a training dataset with 80,000 house sales, and a testing dataset with 20,000 house sales. Next, the model is trained on the training data, and it attempts to learn the relationship between the number of bedrooms, number of bathrooms, and square footage, and the eventual selling price of a house. After training, in preparation for testing, the house prices are removed from the testing dataset so only the features remain. This target-less data is fed to the model, and it attempts to predict the house prices based on the relationship it has learned between the features and the price. Finally, the predictions are compared to the testing dataset's actual house prices to determine the model's accuracy.

There is no objective way to measure model **accuracy**. In this paper, I will use a fairly simple and understandable metric called Mean Average Error (MAE). MAE simply calculates how much, on average, a model's predicted value differs from the actual value of each piece of data in the testing dataset. In the house sales example, a MAE of 5,000 means that, on average, the model's predictions were 5,000 dollars off from the houses' actual selling prices.

Finally, throughout the paper I will refer to two types of models: A neural network and a gradient boosted tree. These are simply two different types of models that have different algorithms for how they attempt to learn the relationship between the features and the target variables. I used both to see if either training algorithm performed better on my specific dataset.

## 2B. Next Gen Stats Tracking Dataset

The Next Gen Stats tracking dataset presents three types of data: data about the context of each game (location, temperature, weather, starting time, etc.), data about the context of each play (starting yard line, in which quarter of the game the play takes place, and various pieces of information about the offensive and defensive formations), and the tracking data itself.

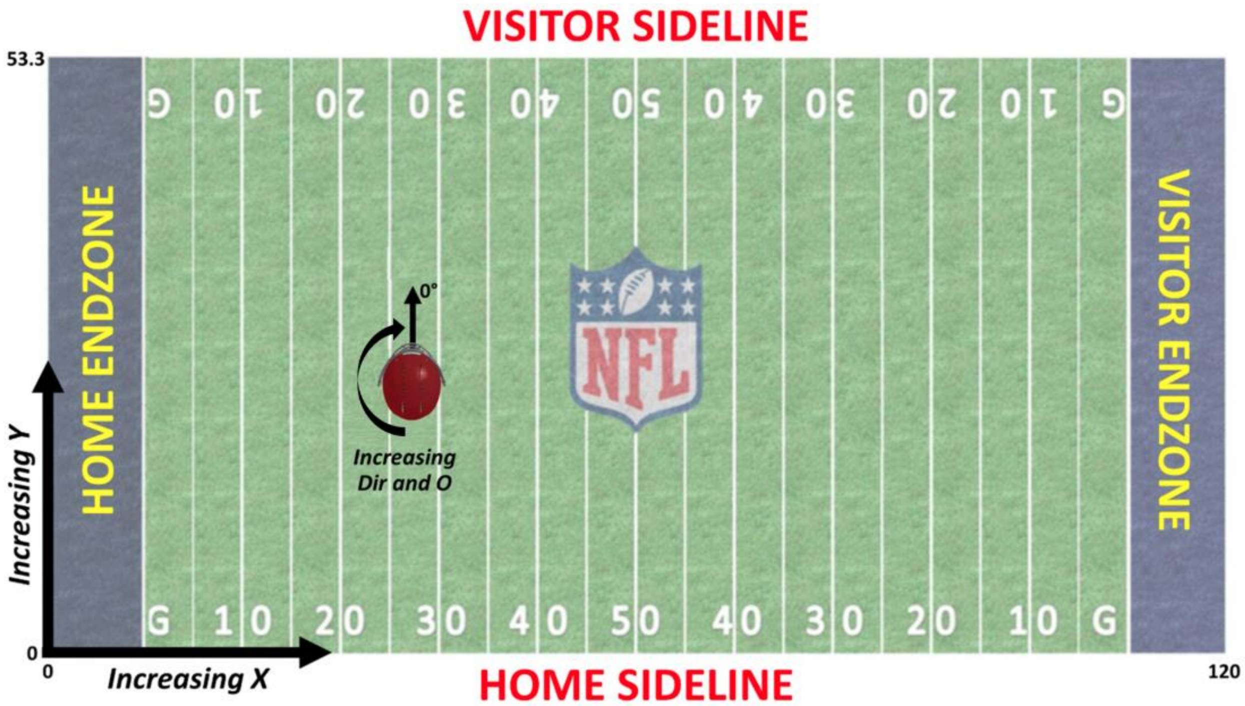


*Figure 1: Visual representation of the Next Gen Stats Tracking Data (Kaggle, 2020)*

The tracking data contains multiple pieces of information about each player. As shown in *Figure 1*, it contains each of the 22 player's X and Y coordinates in relation to the bottom left corner of the field, their direction, given in degrees and using a player facing the visitor sideline as a baseline of 0 degrees, their speed, and the distance they have traveled since the previous timestamp. These pieces of information are recorded at each of many timestamps throughout a play, giving a full picture of each player's positions and movement over the course of an entire play.

When combined, this data gives a very detailed representation of every play of every game. For each game, it gives important context information, and for each play it presents both the context of the play, as well as a detailed description of each player's position and movement throughout the play.

| ***Game Context Information*** | ***Play Context Information*** | ***Player Tracking Information*** |
|---|---|---|
| - *Game Identification*<br>- *Location*<br>- *Weather*<br>- *Temperature*<br>- *Turf Type* | - *Play Identification*<br>- *Quarter*<br>- *Down & Distance*<br>- *Offensive Personnel*<br>- *Defensive Personnel* | - *Play Identification*<br>- *Timestamp Identification*<br>- *X and Y coordinates*<br>- *Direction*<br>- *Speed*<br>- *Distance Travelled* |

*Figure 2: Table showing the types and content of data included in the Next Gen Stats Tracking Data*

The tracking data, however, only records the yards gained from each play. As discussed above, I want to model each play's EPA, not simply its yards gained. Fortunately, as EPA has become so ubiquitous in football analysis, many have created publicly available datasets that contain each play's EPA. I took the EPA data from nflscrapR (Horowitz and Yurko, 2018), a popular data source for EPA that also includes various tools that assist in EPA data analysis. The EPA data from nflscrapR, and the Next Gen Stats tracking data, follow the same game and play identification system, and I was therefore able to match each play in the nflscrapR EPA dataset to its corresponding play in the Next Gen Stats tracking dataset.

## 2C. Data Cleaning

Creating a usable machine learning dataset from the Next Gen Stats tracking data presented the first major challenge for this project. Without going into too much detail, the dataset included every play from the first six weeks of the 2017 regular season, while I only wanted to model the passing plays. Furthermore, it included dozens of timestamps before each play's snap, and after each play's end, resulting in a lot of information that is obviously not relevant to my model. Finally, the data included numerous inconsistencies, and many different aspects of the data had to be manually validated and transformed into usable information. After much data cleaning, I successfully created one consistent dataset comprised of the game, play, and tracking data for every passing play for the first six weeks of the 2017 regular season, limited to timestamps that occurred during the play. To relate this to the modeling overview, the game, play, and tracking data are the features, and the EPA and yards gained are the model's target variables.

## 2D. Feature Engineering

As mentioned previously, prediction models attempt to learn the relationship between features and a target variable. Computers, however, do not have a semantic knowledge and understanding of data. Rather, the relationships learned are purely mathematical. In many cases, therefore, it is very helpful to use the given features to create new features that represent important semantic information. Although it is possible for the model to implicitly learn these features on its own during training, explicitly defining them for the model can greatly increase accuracy. Returning briefly to the fictional house sales dataset: perhaps we suspect that number of bedrooms per number of bathrooms (simply number of bedrooms divided by number of bathrooms) is potentially relevant to the eventual sale price. Without explicitly creating the feature, it is certainly possible that the model will learn this relationship on its own, if it exists. If, however, we compute this value for each house in the dataset and include it as a new feature, the odds of finding a relationship between it and the sale price is increased greatly.

Due to the vast number of features (5 pieces of tracking data for each of 22 on field players, *and* data about each game and play), and a computer's inherent inability to understand the important football semantics that the features represent, it is very important to explicitly define some meaning contained in the data via feature engineering. More specifically, although the data implies various distances between different players, and semantically those distances are very relevant to the outcomes of the plays, they are not explicitly defined. It is certainly possible the model would learn the relationship between the players' coordinates and the success of a play without an explicit definition, but engineering various implied distances in the data increases those odds.

Before explaining the actual features I engineered, it is important to understand the breakdown of the types of players that the offense uses on each play. There are eleven offensive players on the field for every play. One player is always the quarterback, who initiates the play, and is the one who ultimately throws the ball to another player on the team. Out of the other ten players, five are eligible to receive a pass from the quarterback, and five are designated as ineligible to receive a pass from the quarterback. The eligible players are generally comprised of a combination of wide receivers, tight ends, and running backs, while the five ineligible ones are generally comprised of offensive lineman. (I will refer to the first group as skill players, and the latter group as offensive linemen for the remainder of the paper.) Instead of attempting to catch a pass from the quarterback, the offensive linemen attempt to stop the defensive players from tackling the quarterback.

I engineered 4 different types of features, each explicitly defining a distance between a pair of players that I believe has football significance. I believe this explicit definition will aid the model in predicting the outcomes of the plays.

- The distance between the quarterback and the nearest defensive player (Quarterback Pressure). This distance measures how much the defense is affecting the quarterback's throw. A large distance signifies the quarterback is not being pressured by the defenders, while a short distance signifies the opposite.
- The distances between each of the five skill players and the nearest defensive player (Wide Receiver Separation). These distances determine how difficult it will be to successfully complete a pass to that skill player. A large separation provides a much larger margin for error to still

complete the pass, while a small separation creates a very tight window into which the quarterback must throw the ball for a successful completion.

- The distances between the quarterback and each of the five skill players (Wide Receiver Distance). These distances further help quantify how difficult it will be to successfully complete the pass, as longer throws are more difficult to complete.
- The straight-line distances between each of the five skill players and the quarterback (Wide Receiver Depth). This will help quantify how many yards a pass should gain, if successfully completed.

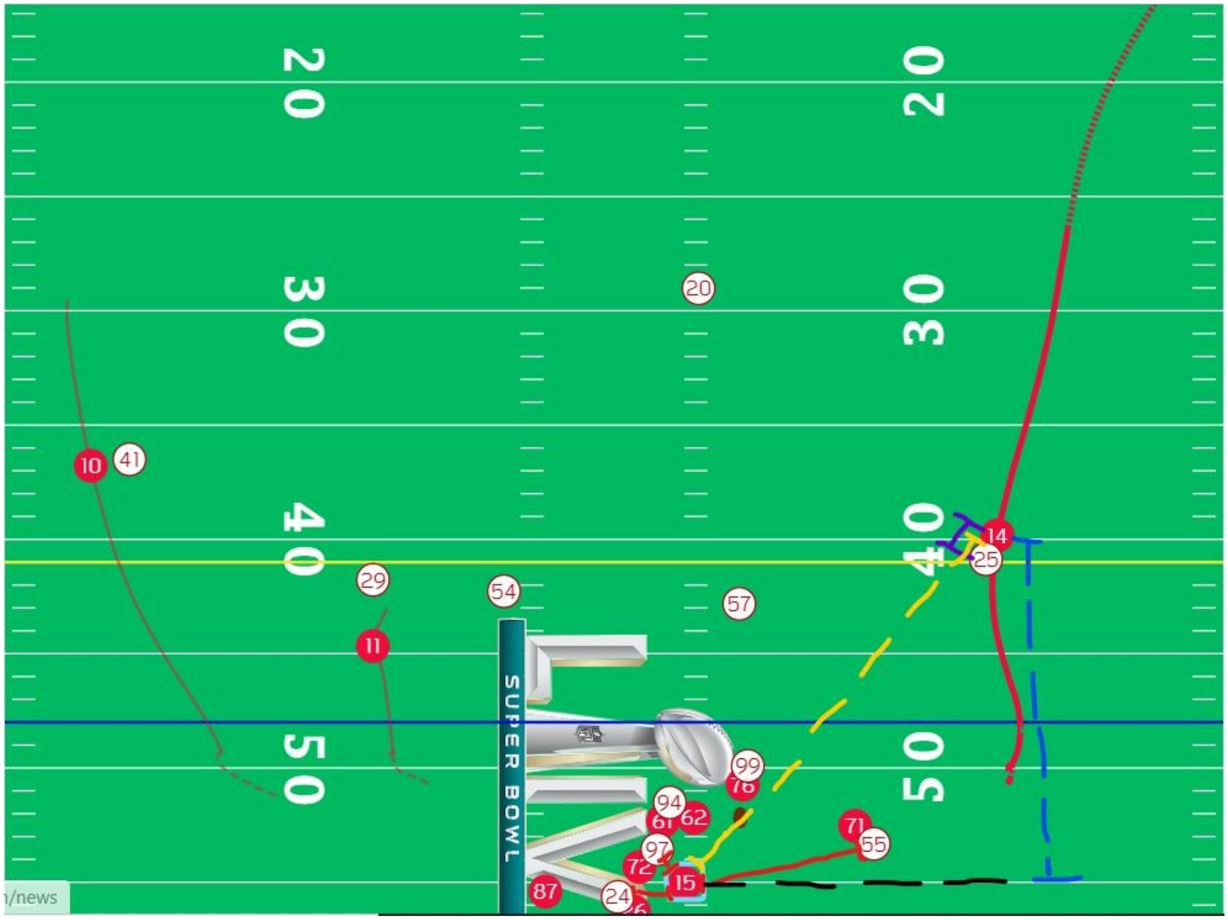


*Figure 3: A visual representation of the four features I engineered, geared at quantifying Quarterback Pressure, Wide Receiver Separation, Wide Receiver Depth, and Wide Receiver Distance (NFL Next Gen Stats, 2020).*

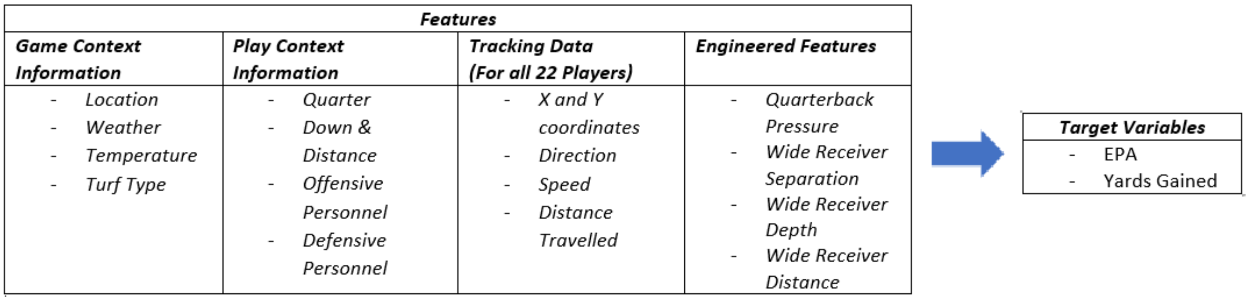
*Figure 3* presents a visual aid for understanding the engineered features, as well as the effect these features could have on an actual play. The image is a dot representation of an NFL play, generated using the Next Gen Stats tracking data, an incredibly useful tool for visualizing football plays. This specific play is taken from this past season's super bowl between the Kansas City Chiefs and the San

Francisco 49ers. The image is the dot representation of the moment the quarterback (Patrick Mahomes, number 15 in red, circled in light blue) threw the ball to the wide receiver (Sammy Watkins, number 14 in red). The features I engineered are drawn on the image, in different colors for clarity.

- Quarterback Pressure: The line drawn between number 97 in white (Nick Bosa) and the quarterback represents the shortest distance between the quarterback and a defensive player. In this specific play, the defensive player is very close to the quarterback, making the play much more difficult.
- Wide Receiver Separation: The line drawn between the wide receiver and number 25 in white (Richard Sherman) represents the shortest distance between the wide receiver and a defensive player. Like the Quarterback Pressure, this distance is very short, making the play much more difficult.
- Wide Receiver Distance: The line drawn between the wide receiver and the quarterback.
- Wide Receiver Depth: The straight-line distance between the wide receiver and the quarterback.

The Wide Receiver Separation, Wide Receiver Distance, and Wide Receiver Depth were calculated separately for each of the five skill players, and thus the feature engineering resulted in 16 new features upon which the model can train.

Thus, after cleaning the dataset, adding the EPA data, and engineering the features, I finally had a complete dataset to train a Machine Learning Model to predict both yards gained and EPA of NFL passing plays.



| *Features* | | | |
|---|---|---|---|
| ***Game Context Information*** | ***Play Context Information*** | ***Tracking Data (For all 22 Players)*** | ***Engineered Features*** |
| - *Location*<br>- *Weather*<br>- *Temperature*<br>- *Turf Type* | - *Quarter*<br>- *Down & Distance*<br>- *Offensive Personnel*<br>- *Defensive Personnel* | - *X and Y coordinates*<br>- *Direction*<br>- *Speed*<br>- *Distance Travelled* | - *Quarterback Pressure*<br>- *Wide Receiver Separation*<br>- *Wide Receiver Depth*<br>- *Wide Receiver Distance* |

| ***Target Variables*** |
|---|
| - EPA<br>- Yards Gained |

*Figure 4: Visual Representation of the dataset's final state. The Features are comprised of the Game Context, The Play Context, The Tracking Data, and the Engineered Features. The Target Variables that we will train the model to predict are the plays' EPA and yards gained.*

After this process of Data Cleaning and Feature Engineering, my dataset contained 195 features for each of approximately 6,500 plays, each of which consisted of, on average, just under 30 timestamps, totaling just over 190,000 rows.

## III. Real Play Example

To better illustrate these features I have engineered, consider the following pictures from an actual NFL game between the Green Bay Packers and the Seattle Seahawks during week one of the 2017 NFL season. This particular play resulted in a 21-yard completion from quarterback Aaron Rodgers to wide receiver Davante Adams. Following the play through a series of pictures perfectly shows the football semantics behind the engineered features.

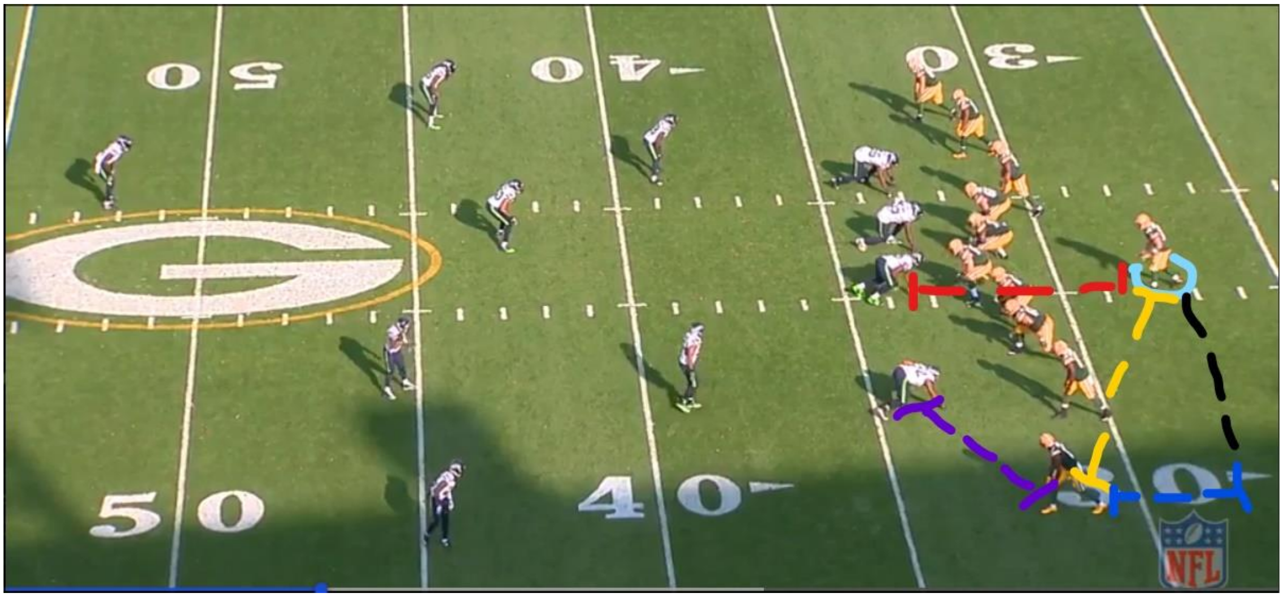


*Figure 5: Picture of the play in question at the snap (NFL All-22 Coach's Film, 2017).*

As seen in *figure 5*, the offense begins the play lined up in a Shotgun formation (the quarterback is standing a few yards behind the offensive line, not directly behind it). The defense is in zone coverage, which means that each defensive player is responsible for defending an area of the field, as opposed to an individual skill player. This will be clear throughout the play, as the nearest defender to Adams (i.e. the defender which determines the Wide Receiver Separation) changes numerous times. At the beginning of the play, Adam's Wide Receiver Separation is 5.96 Yards, his Wide Receiver Distance is 12.53 Yards, and his Wide Receiver Depth is 3.97 Yards. Meanwhile, because Aaron Rodgers is lined up in Shotgun, and is therefore a few yards removed from the Line of Scrimmage, the Quarterback Pressure starts at 5.89 Yards.

A few time stamps into the play (*figure 6*, below), Adams runs down the field directly into a defensive player's zone, and, as the Wide Receiver Depth and Distance rise, the Wide Receiver Separation drops quickly. The defense, however, is unsuccessful in its attempt to pressure Rodgers, and the Quarterback Pressure number rises slightly, signifying that the defense is getting further away from the Quarterback. At this point, the Wide Receiver Separation drops to just .7 yards, the Wide Receiver Distance and Depth grow to 17.4 Yards and 16.18 Yards respectively, and the Quarterback Pressure grows slightly to 6.2 Yards.

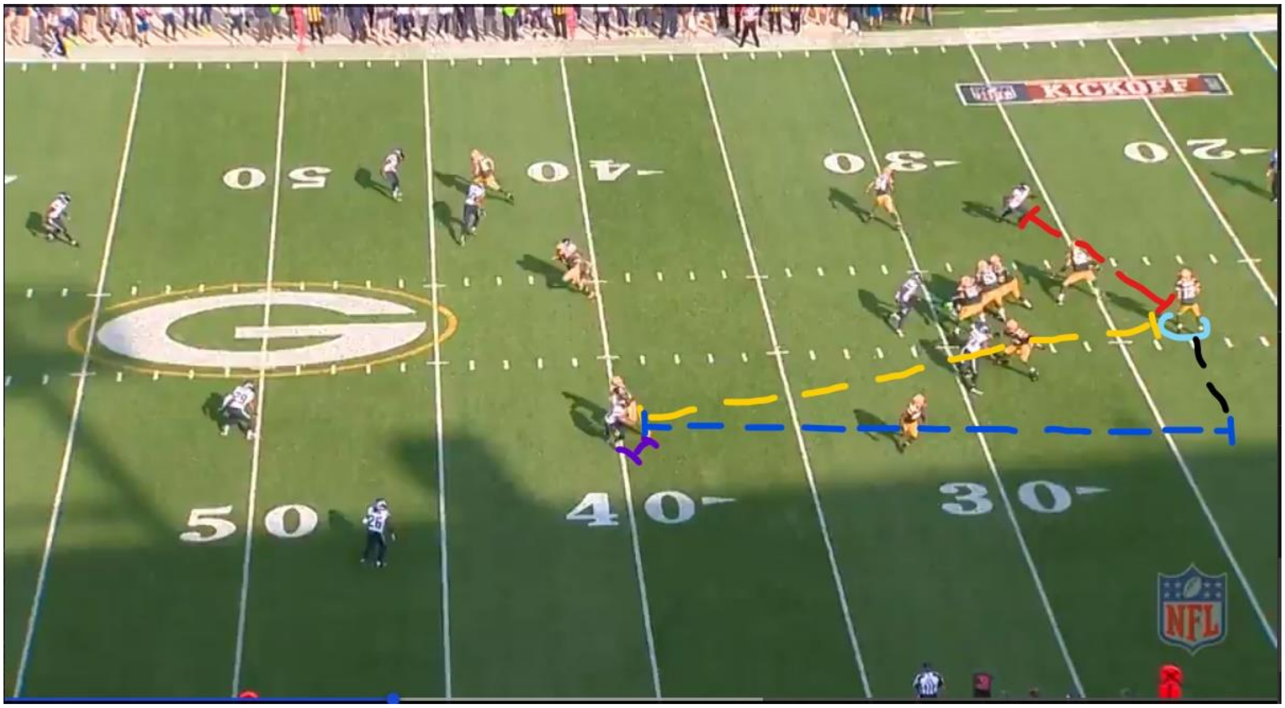


*Figure 6: Picture of the play in question as Adams enters the first defender's zone.*

About halfway through the play (*figure 7,* below), Adams runs further down the field and leaves the initial defender's zone, but is not yet in the next defender's zone. Thus, his Wide Receiver Separation, Depth, and Distance all rise. The defense remains unsuccessful in its attempt to pressure Rodgers, and the Quarterback Pressure remains consistent. At this point, the Wide Receiver Separation jumps to 7.41 yards, the Wide Receiver Distance jumps to 20.68 yards, the Wide Receiver Depth jumps to 20.43 yards, and the Quarterback Pressure drops slightly to 5.61 Yards.

A few seconds later (*Figure 8*, below), Adams enters the next defender's zone, and, once again, the Wide Receiver Separation drops. Meanwhile, the defense finally has some success pressuring Rodgers, and the Quarterback Pressure drops slightly as well. The Wide Receiver Separation falls to 1.7 yards, the Wide Receiver Distance rises to 25.28 yards, the Wide Receiver Depth rises to 25.17 Yards, and the Quarterback pressure drops to 4.34 yards.

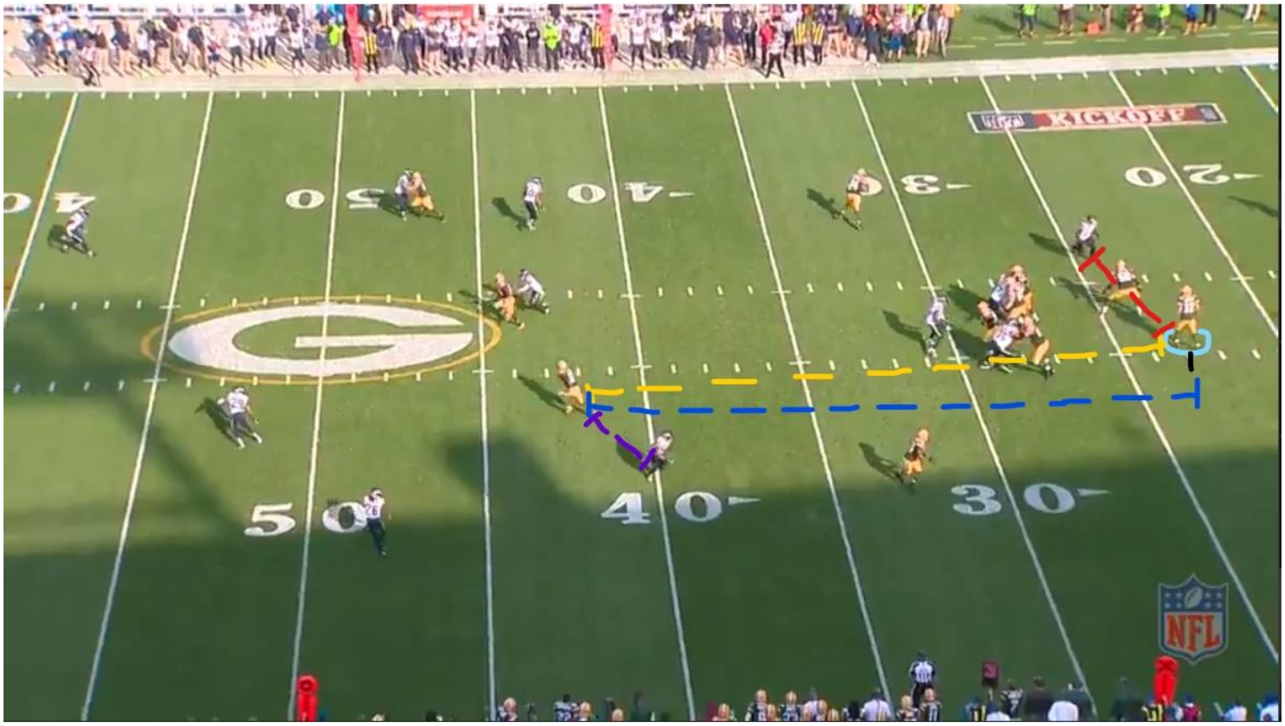


*Figure 7: Picture of the play in question as Adams leaves the first Defender's zone.*

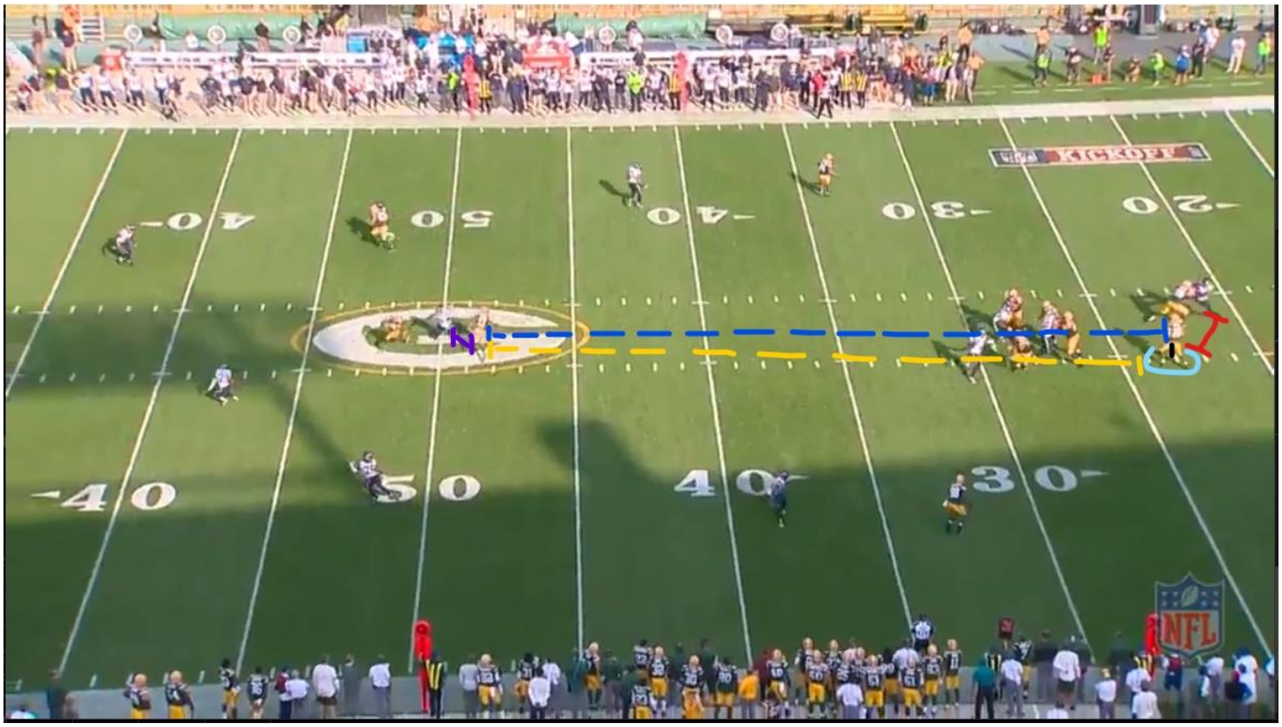


*Figure 8: Picture of the play in question as Adams enters the second defender's zone.*

Although Adams enters a new defender's zone in the previous picture, the defender is already chasing a different Wide Receiver further down the field and fails to notice Adams. This allows Adams to find plenty of open space just a few seconds later as he continues to run across the field (*figure 9*, below). By the time Rodgers releases the ball, Adam's Wide Receiver Separation has skyrocketed. At the release, and the final time stamp the model will consider, Adams has a Wide Receiver Separation of 10.77 Yards, a Wide Receiver Distance of 29.09 yards, and a Wide Receiver Depth of 27.95 yards. Further, the pressure on Rodgers has subsided slightly, as the Quarterback Pressure number jumps to 4.71 yards.

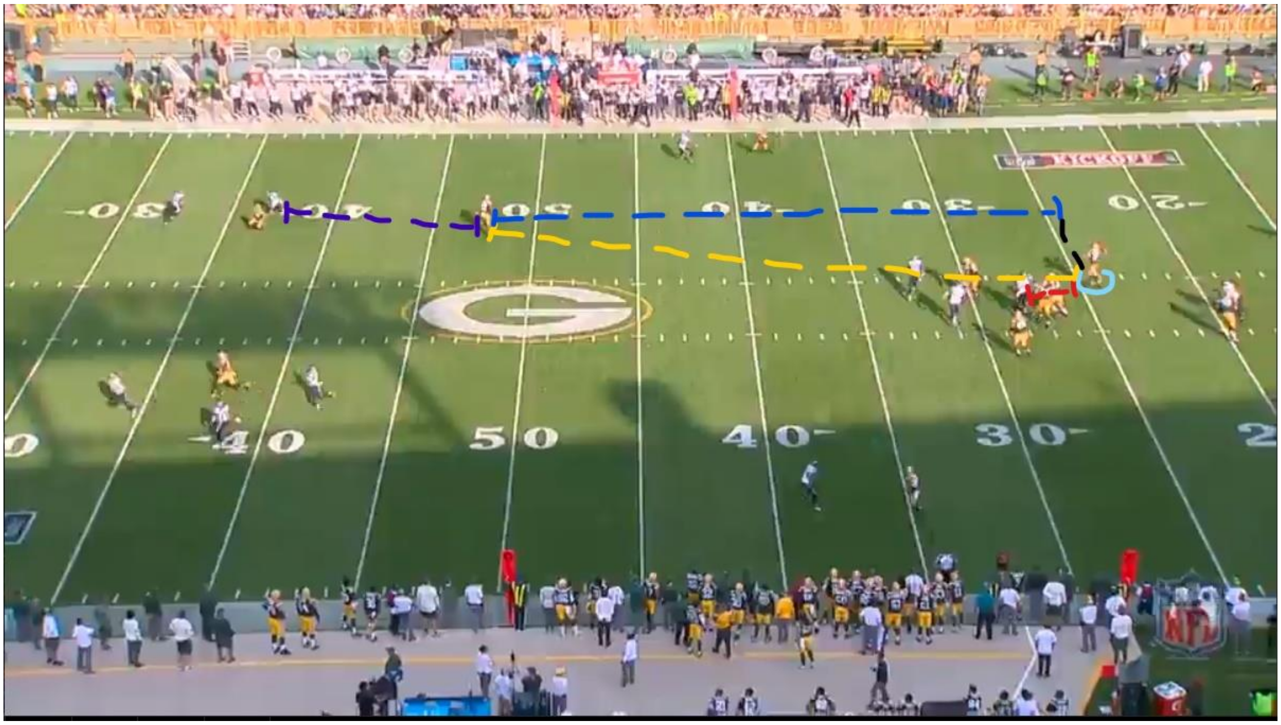


*Figure 9: Picture of the play in question as Rodgers throws the ball. Adams has left the second defender's zone and has opened up across the field.*

Thus, the engineered features, especially when combined, paint a fairly complete picture of a given play, and should give the model valuable information. At the time Rodgers releases the ball on this particular play, the features explain that Adams is about 20 yards down the field with no defensive player in his vicinity, and Rodgers is relatively unbothered by the defensive pressure. Given this information, it is hardly surprising that the play results in a 21-yard completion, and hopefully the model will have an increased chance of learning these football semantics from these explicitly defined features.

# IV. Research Results

## 4A. Definition of Goals & Baseline 'Dummy' Model

As mentioned previously, there is no objective measure for a successful model. Even after choosing an accuracy metric – Mean Average Error – it is still fairly subjective what constitutes a good accuracy score. Is an average error of 5 yards good? Is one of 1.5 Expected Points good?

In order to create some sort of benchmark, I created a "dummy" model and checked its accuracy. Quite simply, I calculated the average yards gained and EPA for each row in the training dataset and used that value as the prediction for each row in the testing dataset. Obviously, this would not even qualify as modeling the data, and is solely to get a benchmark of what bad performance is as a frame of reference for the model's actual results. The averages for yards gained and EPA over the training set are 5.7 and -.4 respectively. When using these numbers as the predictions, they produced MAEs of 7.2 yards and 1.2 Expected Points. My goal for the modeling is to improve these "dummy" results as much as possible.

## 4B. Modeling Results

To predict my target variables most accurately, I ran two separate types of models – A neural network and a gradient boosted tree. Although these models learn quite differently, it is most important to note that they each follow the same general structure outlined in the machine learning overview above. For understanding this paper, therefore, it is not essential to delve into the specifics of how each model processes the data it is fed, but simply that I used two different types of models in hopes of finding the one most suited for accurate prediction of the tracking data.

Ultimately, I was only able to marginally improve over the "dummy" model's accuracy. Even after significant data pre-processing when needed – one hot encoding categorical values, scaling numeric values, removing outliers, etc. – the model's accuracy barely improved. The final modeling results are presented below in *figure 10.*

| ***Model Type*** | ***Target Variable*** | ***MAE*** |
|---|---|---|
| *Neural Network* | *Yards Gained* | *6.1 Yards* |
| *Gradient Boosted Tree* | *Yards Gained* | *6.2 Yards* |
| *Neural Network* | *EPA* | *1.08 Expected Points* |
| *Gradient Boosted Tree* | *EPA* | *1.05 Expected Points* |

*Figure 10: Modeling results*

These results represent just a 15 percent increase in accuracy over the “dummy” model. These are, obviously, not ideal modeling results, and I will investigate why I believe I was unable to improve the models’ accuracies more significantly in the coming sections.

## 4C. Feature Importance

Neural networks make their predictions by computing complex mathematical formulas based off all the features provided, making the model’s final prediction process quite opaque, as these computations end up far too complicated for humans to easily understand. Gradient boosted trees, however, base their predictions off decisions trees, which are significantly more understandable.

*Figure 11* (Data-Flair, 2019) presents a decision tree for whether a person should go to the store to buy shampoo, and it is extremely easy to follow the decision-making process presented. To put this specific decision tree into modeling terms, “Am I out of shampoo?” and “Is it raining?” are the features, and whether the person should go to the market is the target variable. Thus, like all prediction models, decision trees show the relationship between features, and target variables. It is then possible to look at the decision tree and determine which features most significantly impact the final decision, known in modeling terms as feature importance. In this example, both “Is it raining?” and “Am I out of shampoo?” are of equal importance, since the person will go to the market to buy more shampoo if, and only if, they are both out of shampoo *and* it is not raining. In many more complicated decision trees, however, different features will have disproportionate impacts on the final decision/prediction and will therefore have different feature importance scores.

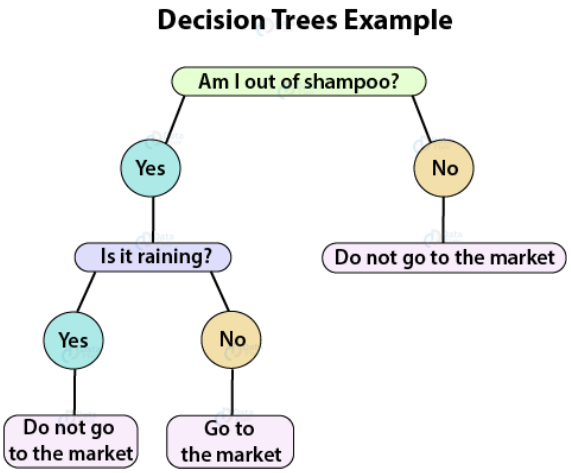


*Figure 11: Visual representation of a Decision Tree*

Although my boosted tree’s accuracy was not ideal, I was still able to retroactively examine the model’s feature importance scores. *Figure 12* (below) shows the top 30 features by F-Score, a common measure of decision tree feature importance, and, despite the poor overall model performance, it does yield many interesting insights that confirm some of my initial thoughts about the data.

First and foremost, the two most important features, by a wide margin, relate to quarterback pressure. The first, “numberOfPassRushers,” one of the defensive personnel features I described as part of the play context feature group, identifies the number of defenders attempting to pressure and tackle the quarterback. The second, “qb_pressure,” is the feature I engineered measuring the distance between the Quarterback and the nearest defender. Next, it is notable that three of the top seven features relate to play context; “MinutesElapsed” counts how many minutes of the game have elapsed, effectively measuring how far into the game the play takes place, “down_3” measures whether the play takes place on third down, and “down_4” measures whether the play takes place on fourth down. These features’ high scores are not surprising, since teams often are more aggressive on later downs and later in the game, which unsurprisingly has an impact on the number of yards that plays in those situations yield. The actual tracking data rounds out the top 10 features (x_3, x_1, x_11, etc. X_3, for example, refers to the X coordinate of the 3rd of the 22 players on the field. The first 11 players on the field are the defensive players, while the remaining ones are the offensive players. More details below in the *Figure 12* description). It is notable that, generally, the defensive players’ coordinates (x_3, x_1, x_11, x_6, x_5, x_2, y_3, y_7, x_7) appear to be impactful than the offensive players’ coordinates. Furthermore, given that the X coordinate refers to the depth of the player down the field (refer to *Figure 1* for a visual aid), it is not surprising that it had a larger impact than the collective players’ Y coordinates.

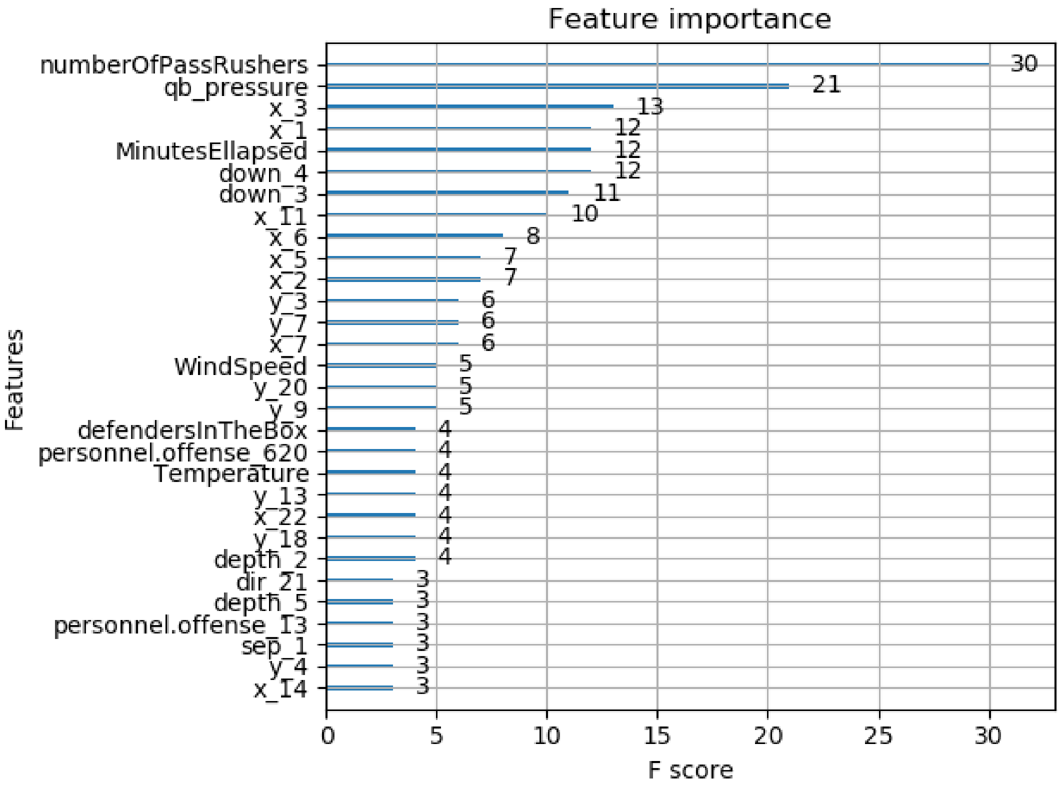


*Figure 12: Feature Importance chart for Gradient Boosted Tree. Top 30 features by F Score. To explain the tracking data feature names: X_1, y_1 through x_11, y_11 are the X and Y coordinates of the defensive players, while x_12, y_12 through x_22, y_22 are the X and Y coordinates of the offensive players. Within the offense, x_12, y_12 through x_16, y_16 are the offensive linemen's coordinates, x_17, y_17 through x_21, y_21 are the skill players' coordinates, and x_22, y_22 are the quarterback's coordinates.*

Finally, many of the other features I engineered appear in this top 30 ranking. Depth_2, and depth_5 refer to the Wide Receiver Depth of the second and fifth Skill Players, respectively, and sep_1 refers to the Wide Receiver Separation of the first skill player.

Ultimately, the neural network and gradient boosted tree models only produced modest improvements over the "dummy" model, both when trying to predict yards gained, and EPA. However, the feature importance chart presents meaningful and interesting information that seems consistent with prior football knowledge and assumptions.

# V. Conclusion & Future Work

## 5A. Reasons for Poor Modeling Accuracy

I believe the main reason for the model's relative inaccuracy was a limited amount of data. Although the dataset totaled nearly 200,000 rows after data cleaning, those rows represented only 6,500 plays, each of which consisted of an average of just under 30 rows. Thus, by modeling standards, each play made up a large percentage of the data. Any inconsistencies on individual plays, therefore, could have a significant negative effect on the model's predictions.

I further believe that the fact that the feature importance scores line up with logical expectations supports this point. The model did successfully pick up on various important football semantics: quarterback pressure, play context, and defensive positions are all significant factors that determine the outcome of a play. Perhaps the fact that the model picked up on these semantics indicates that it did not completely fail to learn a relationship between the features and the targets, but merely did not have enough data on which to learn a totally accurate one.

With more data, and more feature engineering, I therefore believe that a more accurate model is attainable, and I think such a model would have a few significant use cases.

## 5B. Future Work & Use Cases

A model that can accurately predict Yards and/or EPA would have two significant use cases. First, it could be used to quantify NFL coaching. As is, no real method exists to quantify a coach's performance, as it is definitionally intertwined with the players' performances. With such a model, however, analysts would be able to determine how often teams are in positions to succeed, and slightly decouple the play's eventual result with the expected result. A coach whose team is consistently in a position for which the model predicts good performance is likely putting the players in a position to succeed. Although this is still considerably coupled with the players' performances, it would at least decouple a coach's performance from the plays' eventual results.

Second, and more significantly, an accurate model would allow for unparalleled player performance evaluation, specifically for quarterbacks. As mentioned above, EPA has become the gold standard for team, and player, evaluation, because it places plays from very different contexts onto the same scale of success and failure. This allows analysts to compare all plays, regardless of the context in which they happened. In the current football analytics world, a commonly used measure of performance is EPA per play (EPA/P), which measure how successful, on average, is a set of plays. On a team level, analysts will use EPA/P to determine teams' average offensive and defensive success. On an individual level, analysts will use the EPA/P of all plays in which an individual player is involved to measure that player's performance. EPA/P is most commonly used as an evaluation metric for quarterbacks, and many will use one quarterback's higher EPA/P than another's to claim that quarterback is the better player of the two.

Obviously, there is a lot of validity to that analysis. After all, it does measure the average success of plays in which the quarterbacks are involved. However, although EPA (and in turn EPA/P) adjusts for game context, it does not adjust for play context. Although EPA measures how successful a play was, it does not measure how much more/less successful a play *was* than what it *should have been*. Consider the following two fictional plays:

1. 1st and Goal, on the opponent's 1 yard-line. The quarterback miraculously avoids multiple defenders, and, despite none of his skill players having any Wide Receiver Separation, manages to complete the pass for a 1-yard Touchdown.
2. 4th and 10, from the offense's own 1 yard-line. The quarterback throws a short, wide open pass to a wide receiver, who then runs for an easy 99-yard Touchdown due to the defense's ineptitude.

Neither traditional football statistics nor EPA capture the true value of either of these plays. Traditional statistics would simply record the former as a 1-yard touchdown and the latter as a 99-yard touchdown. EPA, on the other hand, would capture the actual value of the plays, but miss the important in-play context. Recall that EPA is based off a statistical prediction of the difference in how many points the offense's drive should net before and after a play. The first play would, therefore, have its value decreased, since it was extremely likely going into the play that the offense would finish the drive with a touchdown. Actually scoring a touchdown does not then significantly increase the expected value of the drive. The second play, on the other hand, would have its value dramatically increased. Since it is extremely unlikely the offense scores any points in that game situation, actually scoring a touchdown greatly increases the expected value of the drive. This, however, is problematic for analysts who intend to use EPA/P as a quarterback evaluation metric, as, from a quarterback performance perspective, the first play, which has a low EPA is a result of wildly impressive quarterback play, while the second play, which has a very high EPA, is a result of factors outside of the quarterback's control.

This touches upon a more general issue in creating insightful football statistics. An insightful, predictive statistic boils down to accurately attributing success and failure to each individual player, as that allows analysts to predict individual players' performances in future situations. Doing so for football presents two separate difficulties. First, each play has its own, unique goal. A 10-yard pass on 3rd and 9 and a 10-yard pass on 4th and 11, although identical on the traditions stat sheet, represent opposite levels of success. The former is a very successful play, while the latter is a complete failure. Second, each play has 22 moving players, making it almost impossible to accurately attribute credit and blame to each individual player. If a quarterback throws a touchdown, for example, it is almost impossible to determine whether the quarterback made a good play, the wide receiver made a good play, or a defensive player made a bad play.

By adjusting for *game* context, EPA fixes the first of these two issues, as it maps each play onto a scale of success and failure instead of one of yards gained. EPA, however, fails to adjust for *play* context, as it does not attempt to accurately attribute credit and blame to individual players for successful and unsuccessful plays. It is, therefore, problematic to solely rely on EPA/P when evaluating individual players' performances. If a quarterback has an EPA/P of .5, although some will use that as an objective measure of the Quarterback's success, in reality it simply means that plays in which the quarterback was involved averaged a gain of .5 Expected Points. It does not, however, indicate what amount of those .5 points are because of the quarterback's play, and the amount that are due to other factors.

This is how an accurate model would greatly improve the world of quarterback evaluation. It would, to an extent, enable analysts to adjust plays not just for their *game* context (as EPA currently does), but also for their *play* context. Instead of simply using a quarterback's EPA/P, it would enable analysts to determine how many more/fewer Expected Points each of the quarterback's plays produced compared to the model's predictions. To illustrate, consider again the two fictional plays outlined above. A model that can successfully predict EPA would fix the issue left by traditional EPA. Because of the immense Quarterback Pressure and the limited Wide Receiver Separation, the model would presumably predict a bad outcome for the offense in the first play. Conversely, because of the large Wide Receiver Separation and the inept defensive performance, the model would presumably predict a huge success for the offense in the second play. When comparing those projections to the actual result, the first play, because its result outperformed expectations, would be rewarded, while the second play, because its result simply fulfilled expectations, would be penalized.

Thus, such a model would enable analysts to create a new statistic – EPA Over Expected (EPAOE) – that would more accurately capture each play's in-play context, and in turn each quarterback's actual level of success. It would judge, for each play, how many Expected Points *should* be gained by the offense, and will judge quarterback performance by whether their plays over or underperform these expectations. (Additionally, because the model will be trained on league-wide data, it is definitionally scaled to league average. This allows analysts to view EPAOE as a statistic measuring quarterback performance compared to what a league average quarterback could have done) Although EPAOE would obviously not be able to account for every factor out of the quarterback's control, it would still provide a significant step in the right direction over traditional EPA/P.

## 5C. Conclusion

In this paper, I have provided a detailed introduction to modeling NFL Next Gen Stats Tracking Data to tackle complicated football analytics problems. I have further outlined why I believe such modeling is important, and how it can add to the current world of football analytics. Finally, I attempted to use the tracking data to model both EPA and yards gained, and, although ultimately not as successful as I had hoped, I believe I showed that there is reason to believe that the data can be modeled successfully with more data and more advanced feature engineering.

# Bibliography

Next Gen Stats, National Football League Operations, 2015, URL: https://operations.nfl.com/the-game/technology/nfl-next-gen-stats/

Big Data Bowl 2019, National Football League Operations, 2019, URL: https://operations.nfl.com/the-game/big-data-bowl/2019-big-data-bowl/

Big Data Bowl, National Football League Operations, 2019, URL: https://operations.nfl.com/the-game/big-data-bowl/

Alok Pattani, ESPN Analytics, 2012, URL:https://www.espn.com/nfl/story/_/id/8379024/nfl-explaining-expected-points-metric

Maksim Horowitz, Ron Yurko, NFLScrapR, 2018, URL: https://github.com/ryurko/nflscrapR-data/blob/master/legacy_data/README.md

Big Data Bowl 2020, Kaggle Competition, 202, URL: https://www.kaggle.com/c/nfl-big-data-bowl-2020/data

Big Play Highlights, Next Gen Stats, National Football League, 2020, URL: https://nextgenstats.nfl.com/highlights/play/type/team/season/week/playerId/2020020200/3406

All-22 Coach’s Film, National Football League, 2017, URL: https://gamepass.nfl.com/game/seahawks-at-packers-on-09102017?coach=true

Data-Flair, R Decision Trees – The Best Tutorial on Tree Based Modeling in R!, 2019, URL: https://data-flair.training/blogs/r-decision-trees/