THE OHIO STATE UNIVERSITY

Data Analysis of Pivotal American Football Plays

Project Category: NFL Dataset
Physics 5680, Autumn 2024

Author: Zavier Kamath

July 8, 2025

Abstract

Many people are shocked when watching the pivotal plays in a game of football. It may be difficult to understand the causes, correlations, and impacts of pivotal plays. This paper serves as a full summary of how Python data analytics (specifically keras, and scikit-learn algorithms) was used to understand and predict pivotal plays in football games. First, the correlation of fumbles and other features was found using the feature importance results of random forest algorithms. It was concluded that fumbles are most likely to occur just after a player catches the ball, and when there is less time remaining in the game. Next, a neural network model was trained to predict if a play has a shotgun formation or not with 76% accuracy.

1 Introduction

While watching football, it is easy to notice certain trends that can have big impacts on the game. For example, one may notice specific player formations, turnovers, injuries, and penalties and think to themselves that if that play did not happen, the outcome of the game would have been different. While spending time with avid football watchers, I often hear them talk about this. They say things like "if only he didn't get injured" or "if only he didn't fumble". Hearing these things motivated me to try to uncover the actual truth of the pivotal moments in football. Why do these pivotal plays happen, and what impact do they truly have? These plays are also important for sports analysts to understand due to their potential impact on the outcome of a game.

To understand why these plays happen and what impact they have, we first must understand some key terms. I will assume that the reader already has basic understanding of the game of football. Here, I will only explain the specialty terms that are necessary to understand for this paper:

- Fumble - When a player loses possession of the ball from their arms and possession is turned over to the other team. This is one type of turnover, but interceptions (where a player on the opposing team catches a ball that is being passed to a player on the main team) are another type.
- Shotgun - A type of play where the quarterback stands far back from the line of scrimmage, giving them a better view of the defending team's setup and allowing for a wider variety of plays.
- WPA (Win Percentage Added) - The amount of win likelihood added as a percent due to the result of a play. Used to quantify how good or bad a play was.

- Score differential - The difference in offense team vs defense team score

In this paper we specifically focus on the impact and cause of shotgun formation plays and fumble plays. To do this, our models (Random Forest in the case of the fumble predictor model and Neural Network in the case of the shotgun predictor model) take in various numerical input features of a given play ranging from the time remaining in a game, yards gained, pass length, etc. and output whether they determine a fumble (or no fumble) or shotgun (or no shotgun) will occur. This is a form of binary classification where the outcome is one of two options. Python libraries scikit-learn and keras were used for the machine learning algorithms for the entirety of this project.

How do we know that shotgun formation plays and fumble plays are pivotal? We can look at the average WPA for pass plays where there was a fumble and where there was no fumble:

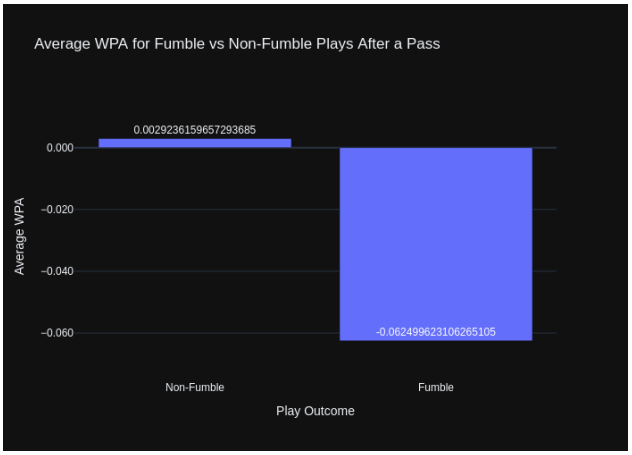

Figure 1: Vizualization of Average WPA for Fumble vs Non-Fumble Plays. This figure shows that plays containing a fumble have a large negative win percentage added, and plays without a fumble have a small positive win percentage added.

And similarly we can notice the same trend for shotgun formation plays and non shotgun formation plays:

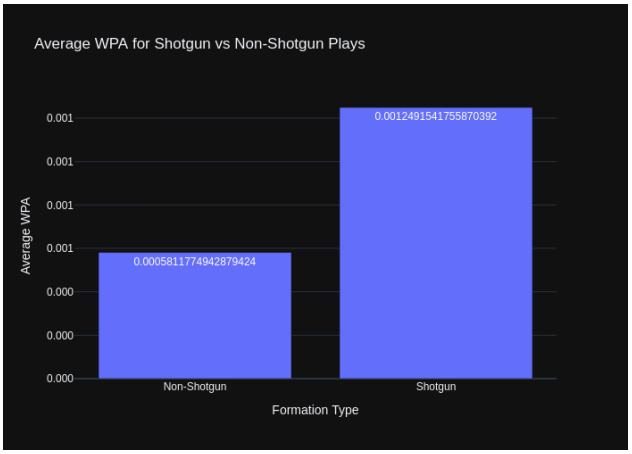

Figure 2: Vizualization of Average WPA for Shotgun vs Non-Shotgun Plays. We can see that shotgun plays have a larger win percentage added thhan non-shotgun plays.

Fumbling significantly decreases the chance of winning any given football game. Shotgun formation plays on average increase the chance of winning.

2 Related Work

In Ref. [1], multiple machine learning algorithms were used to predict play type (run or pass) for various teams during various games. The algorithms used were logistic regression, linear discriminant analysis, gradient boosting machine, and random forest. Logistic regression and linear discriminant analysis methods were not as effective, but gradient boosting (something that combines decision trees) and random forest proved most effective. This study also concluded that certain teams could be predicted extremely accurately on certain days, while other teams were extremely unpredictable on certain days. Overall, they were able to get a model that could predict play type with 75.9% accuracy. Their random forest methods and conclusions are very similar to what I obtain from my neural network modeling of shotgun plays.

Ref. [2] also tested multiple machine learning models to predict play type (run or pass). The difference here is that they also tested neural networks, and they concluded that neural networks had the best performance (even when compared to random forest). This study also went into deeper detail to see how good their performance could be improved when looking at specific scenarios, such as only looking at plays after the team had no huddle. They obtained varying results, but certain scenarios were able to significantly improve predictability. Due to the overall better performance of neural networks in this study, I decided to use a neural network as my shotgun formation prediction model.

Lastly, Ref. [3] used gradient boosting models to predict turnovers (such as fumbles). They found false discovery rates (prediction of a turnover when there really was no turnover) less than 0.15 for 62.5% of teams. This provides me a benchmark to compare my results to for my fumble prediction modeling.

Note that in this project, I am specifically analyzing fumble plays instead of the broader classification of turnover plays. I also create a model to predict shotgun formation occurrence, whereas most of the related works in this field create models to predict the broader classification of play type (run or pass). Looking at these specific pivotal plays may provide better accuracy or interesting results compared to the results of related works.

3 Dataset

The dataset used throughout this project is the regular season play-by-play data for National Football League (NFL) games from 2009 to 2019 [4]. This dataset has a total of 498393 plays, with each play representing a row in the dataset. Each row has 256 features ranging from play id to WPA and many others. Any kind of information about a football play that you might want to know about is encoded in one of these features in some way. There is information about the type of play, where the play happened, who performed the play, when the play happened, etc. Depending on the type of analysis being performed, different features could be selected or dropped.

In order to create a model that can predict if a fumble occurs, I only selected 10 features. These special features are: yards after catch, quarter seconds remaining, half seconds remaining, game seconds remaining, score differential, pass location (left, middle, or right), pass length, down (1st, 2nd, ...), air yards (amount of time the ball spent in the air), and yards gained. I chose these features because they rarely have empty data entries and most of these features have numerical data or something that could easily be turned into numerical data. I also only considered plays where a pass occurred and was completed. This way, we can analyze specifically the case of fumbling occurring during pass plays, which could have different causes and correlations than the case of fumbling during a rush play. I normalized all features to be values between -1 and 1. This included changing the pass location feature to have a value of -1 for right, 0 for middle, and 1 for left. I did not include plays that contained empty data. Lastly, I shuffled the data frame rows. With these changes made, the dataset now only has 120871 plays (about 24% of the original full dataset). During the model training, k-fold validation was used with 20% of the data was designated for validation, while the remaining 80% was used for training during each k-fold iteration.

Here is a printout of what some of the columns of a single play's data can look like:

play	yards_after_catch	quarter_seconds_remaining	score_differential	pass_location	air_yards	...
1	0.175	-0.582	-0.148	-1.0	-0.197	...

To create a model to predict if a shotgun formation occurs, 15 features were selected. These features were down, yards to go (for a 1st down), yard line, goal to go (which signifies if the team in play is within 10 yards of getting a touchdown). game seconds remaining, score differential, timeouts remaining for the team in possession and the defending team, home team name, away team name, team in possession, team defending, huddle occurrence, game half, and win percentage of team in possession. Once again, these features were selected because they are either numerical data, or have the ability to be easily turned into numerical data with a majority of entries being non-empty. Once again, rows with empty entries were dropped, and the data frame was shuffled. This time, features were normalized to be between zero and one. For any feature that included a team name, these columns were one-hot encoded to contain boolean entries. The data was split into 80% train and 20% test, with 20% of the training data being designated to validation.

Here is a printout of what some of the columns of a single play's data can look like:

play	yds_to_go	wp	score_differential	posteam_timeouts_remaining	defteam_OAK	...
1	0.337	0.145	0.1156	0.597	False	...

4 Methods

The machine learning algorithms used to address these problems are:

- A Fully Connected Neural Network for the analysis of shotgun formation plays.
- A Random Forest Regressor for the analysis of fumble plays

4.1 Method for Shotgun Formation Analysis

After the input data has been normalized, it is input into a fully connected neural network. A fully connected neural network is one that takes in an input in the form of a list of numbers, passes those numbers to a layer of "neurons" that perform matrix calculations on them to find hidden patterns, which then pass the new numbers to the next layer that performs more matrix calculations, which eventually end up in the final layer that outputs the result. Along every step of the way, the matrices used to perform the calculations are saved. At the end of this process called the "forward pass", the performance of the network is calculated, and depending on how good or bad the network performed, the values in the matrices used to perform calculations within the network are changed.

Each layer in the neural network has an activation function, used to ensure that the values being computed on are always within a normalized range. I used the relu function for all layers except the last one, where I used sigmoid. Relu is a simple function that generalizes the patterns found by the neural network, which helps prevent over fitting. Sigmoid is used in the final step to give a value between 0 and 1, which can be easily translated into a probability so that the neural network can easily classify the output into the class that has the highest probability.

The neural network I used takes in the normalized input of numbers, passes these numbers to a layer of 64 neurons, which passes to another layer of 32 neurons, which passes to the output layer. This shape of 64 to 32 neurons is useful because it allows the model to find complex patterns with the large 64 neuron layer, and then generalizes those patters in the 32 neuron layer, which prevents over fitting.

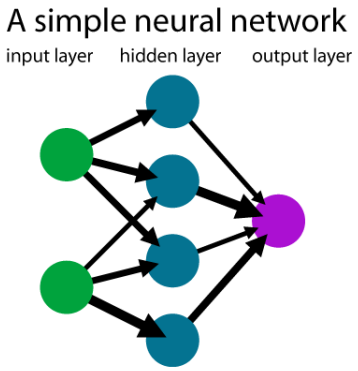

Figure 3: Simple Neural Network Visualization Ref. [5]. This image shows a simple neural network with an input layer, hidden layer, and output layer. This is similar to the neural network used in this project, except there are multiple hidden layers, and there are significantly more neurons per layer.

4.2 Method for Fumble Play Analysis

To analyze fumble plays, I used a random forest model. Random forests are collections of decision trees. Decision trees are models that classify data entries into one class or another through a series of decisions. Each decision is made by seeing if a data entry is above or below a cut value. The value of the cut is determined by finding the value that minimizes the gini impurity, which is the probability of falsely labeling a sample.

As previously mentioned, random forests are collections of decision trees. Random forests make up for the over fitting tendencies of decision trees by training each tree on only a fraction of the overall sample (bootstrapping). This allows each tree to see similar, but not identical data. The results of every tree are then averaged in a process called "aggregation".

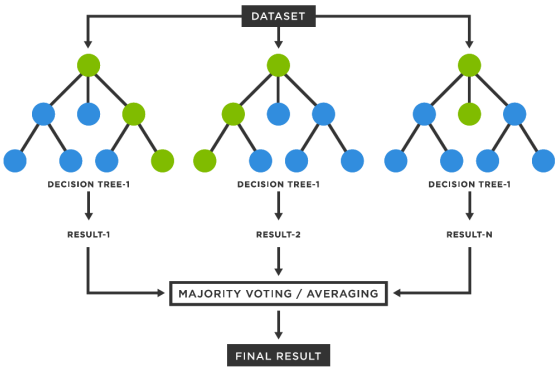

Figure 4: Simple Random Forest Algorithm Visualization Ref. [6]. This image shows a simple random forest algorithm with three different trees. Each tree makes a series of decisions and then outputs what it thinks is the correct answer. There answers go through a process to determine the true correct answer (usually through taking each answer and weighing it by its corresponding confidence probability).

In the case of the fumble dataset, each sample has 10 numerical features and 1 label, which are passed to the random forest model. The model then outputs whether the sample is more likely to be a fumble play or not a fumble play.

One advantage of random forest models is that they can provide us with feature importance information, so that we can deduce which features were the most important for determining if a fumble occurs. I used

linear regression on some of the key features to determine how exactly the feature scales with fumbling (is it a positive or negative relationship). Linear regression is the process of finding a linear best-fit line that best represents the relationship between the feature and fumbling. This is done by minimizing the squared difference between the closest point on the arbitrary line of best fit and the real data points (a process called least squared error fitting).

5 Results/Discussion

Each project (fumble analysis and shotgun analysis) has its own results to discuss, so I will split this section in two.

5.1 Results for Shotgun Formation Analysis

As stated in the methods section, I used a fully connected neural network to analyze shotgun formation plays. Here are the hyper-parameters that I used to fit the model described in the methods section to the data described in the dataset section:

- Maximum of 40 epochs
- Batch size of 32 plays
- Validation split of 20%
- Patience of 6 tracking the validation loss (when the validation loss hits a minimum, the fitting will only go for 6 more epochs before stopping unless it encounters a new validation loss low)

40 epochs were used to allow sufficient time for the model to reach a low validation loss. A batch size of 32 plays allows the fitter to fit quickly. After fitting the model, the best model was saved and applied to the testing data. As mentioned in the methods section, a training-testing split of 80%-20% was used. This provides a huge amount of data to train with while also allowing a significant amount to obtain metrics with. A patience of 6 epochs was used to prevent overfitting.

	Predicted: No Shotgun	Predicted: Shotgun
Actual: No Shotgun	35701	7889
Actual: Shotgun	11868	28955

Table 1: Confusion matrix for shotgun play analysis. Shows true vs predicted shotgun formation plays and non-shotgun formation plays.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Non-Shotgun	0.75	0.82	0.78	43590
Shotgun	0.79	0.71	0.75	40823
Accuracy		0.77		84413
Macro avg	0.77	0.76	0.76	84413
Weighted avg	0.77	0.77	0.77	84413

Table 2: Classification report for the model's performance on the shotgun play prediction task. Shows precision, recall, F1-score, and Support values for each class (shotgun or non-shotgun), as well as the accuracy, macro and weighted averages of the the classes combined.

The main interpretation of these results is that shotgun plays are predictable. The model can predict shotgun plays with a 79% precision and 77% accuracy. While the confusion matrix shows that not all predictions are correct, there is a general trend that we can predict a shotgun (or non-shotgun) formation play correctly much more often than incorrectly.

5.2 Results for Fumble Play Analysis

As stated in the methods section, I used a Random Forest model to analyze shotgun formation plays. Here are the hyper-parameters that I used to fit the model described in the methods section to the data described in the dataset section:

- 100 Decision Trees
- Max Depth of 6 (Limits how complex the tree can be)
- 5 k-folds (a k-fold is when a portion of the data is selected to be used as validation and the remainder is used for training, then this process repeats with another portion of the data until all of the data has been used. The metric results are collected and averaged)

To find the number of decision trees and max depth value that maximized the performance of the model on the testing data, I plotted the performance for a set of each parameter. The most optimal values ended up being a max depth of 4 and 100 decision trees. A max depth of 4 prevents overfitting while still allowing the model to find trends. Having 100 trees allows the model to average the results of many trees without having an over-reliance or under-reliance on any set of trees. 5 k-folds were used to prevent overfitting to a given subset of training data.

	Predicted: Non-Fumble	Predicted: Fumble
Actual: Non-Fumble	20951	2947
Actual: Fumble	201	75

Table 3: Confusion matrix for fumble play analysis. Shows true vs predicted fumble formation plays and non-fumble formation plays.

Metric	Train	Test
Precision	0.992	0.990
Recall	0.879	0.877
AUC	0.721	0.631

Table 4: Model performance metrics for the fumble play prediction task. Metrics include precision, recall, and area under the ROC curve (AUC) for both training and test datasets.

Because non-fumble plays are extremely common compared to fumble plays, the model was trained to identify 85% of plays as non-fumble. This results in many false predictions, but overall the model performs very well. Although the AUC score is not great, the precision and recall scores are fairly high. We can also see from the confusion matrix that the model is not just simply predicting non-fumble for every single play (which is a method that would likely give decent results).

I also ran feature importance diagnostics on the random forest model to determine which features had the most importance. After finding these important features, I ran a simple linear regressor to determine to relationship between the features and fumbling. It was found that players are more likely to fumble the ball soon after catching the ball (this was the #1 importance feature) and that players are more likely to fumble the ball when there is less time in the game (this was the #2 importance feature). This could be because when a player catches the ball, it takes time for them to secure it in their arms, making them more likely to fumble. Also, players are often under further stress towards the end of the game, making them more likely to make mistakes (like fumbling).

6 Conclusions/Future Work

Determining the importance and predictability of pivotal plays in football is important for sports analytics and for understanding and enjoying the game of football in general.

Fumbles, because they are very uncommon, are hard to predict. Despite this, our random forest regressor model was able to predict when a fumble does not occur very accurately. Shotgun plays, on the other hand, are very common. Our fully connected neural network model was able to predict when this occurs with a 77% accuracy.

It is possible that the models and hyper-parameters chosen in this analysis were not ideal for the task at hand. Further work could be done with different models and hyper-parameters to obtain better results.

Beyond this, there are other pivotal plays in football that could be analyzed (such as touchdowns, sacks, field goals, etc...) using similar methods to the ones used in this paper.

References

[1] P. Lee, R. Chen, and V. Lakshman, *Predicting offensive play types in the national football league*, 2016.

[2] C. Joash Fernandes, R. Yakubov, Y. Li, A. K. Prasad, and T. C. Chan, *Predicting plays in the national football league*, *Journal of Sports Analytics* **6** (Feb, 2020) 35–43. Published: 27 February 2020.

[3] J. R. Bock, *Empirical prediction of turnovers in nfl football*, *Sports* **5** (2017), no. 1.

[4] ryurko, "Repository for nfl data." `https://github.com/ryurko/nflscrapR-data/tree/master`, 2019.

[5] H. Hsu, *How do neural network systems work?*, Aug, 2020.

[6] Spotfire, "What is a random forest?."