Predicting Play Outcomes in the NFL

1st Tucker, Taylor
University of Virginia School of Data Science
Charlottesville, United States
rtt4fb@virginia.edu

2nd Shoriz, George
University of Virginia School of Data Science
Charlottesville, United States
pvq8hv@virginia.edu

3rd Lisman, Zack
University of Virginia School of Data Science
Charlottesville, United States
zjl2ue@virginia.edu

4th Casillas-Colon, Abner
University of Virginia School of Data Science
Charlottesville, United States
aec4hr@virginia.edu

5th Lotane, Charles
University of Virginia School of Data Science
Charlottesville, United States
aec4hr@virginia.edu

***Abstract*—This study focuses on the convergence of machine learning and American Football, aiming to predict play outcomes using machine learning techniques to provide actionable insights for team strategy and player performance. Our methodology consists of three primary experiments: a sequence model utilizing recurrent neural networks (RNNs) to predict subsequent plays, a fully connected neural network to estimate play distance, and a binary classification model to distinguish between pass and run plays. The sequence model achieved a training accuracy of 54.74%, indicating moderate predictive power. The play distance model significantly enhanced play prediction accuracy, achieving 75%, while the binary classification model successfully differentiated play types with a validation accuracy of 72.78%. The findings of this study have important implications for NFL teams looking to optimize offensive and defensive strategies, while also providing improved analytical tools for fans and stakeholders interested in the game's changing landscape.**

***Index Terms*—Data Science, Machine Learning, Artificial Intelligence, NFL, Predictive AI**

I. Introduction

With AWS and Amazon becoming more involved in the NFL than ever before, there is a statistical revolution occurring in football. Stats such as catch probability, separation distance, win probability, and many others have been introduced to football and are completely changing the way teams play the game.

In this study, we will be using our knowledge of machine learning and neural networks to attempt to predict outcomes of NFL plays. These outcomes include how many yards are expected to be gained or lost on a play, whether the play was a run, a pass, or a trick play, and how expected points and win probability are affected on a play. This is important for NFL teams because they would like to be able to run plays on offense that will gain lots of yards and points, and run plays on defense that will prevent yards and points. If a team knows that a particular play or formation works well on either the offensive or defensive side of the ball, it gives them a distinct advantage. Coaches and teams watch a lot of film to help them prepare for the games, but if we can take still images and videos and help provide insight on each play in addition to what is shown, it can add more helpful insight to teams.

Additionally, this analysis can be very useful to help analyze individual players. If a player performs a negative action such as dropping a pass or missing a tackle, or a positive action such as making a tough catch or covering a lot of ground to break up a pass, the impact of that player on winning can be more accurately quantified.

II. Motivation

As mentioned during the introduction, there is a large amount of interest for the ability to predict plays in the NFL. The ability to effectively evaluate the problem of play efficiency is an important problem at both the micro and macro level. Predicting plays alone is a problem that provides value in the NFL for teams and spectators who would want to judge how likely a pay is to succeed. This gives teams more tools to adjust accordingly against their opponents and fans more knowledge about the intricacies of the game even with less understanding on the technical elements of unique player positions. There are additional benefits of modeling this problem that can be applied to larger meta analyses. Some examples of how these are modeling win likelihood, evaluating the efficiency of plays in general to improve scheme, modeling player value by controlling for play effectiveness ,or in predicting outcomes for sports betting.

III. Dataset

Our dataset was compiled by a statistical research team from Carnegie Mellon University comprised of Ron Yurko, Sam Ventura, and Maksim Horowitz, who also developed `nflscrapR`, an R package which leverages the NFL's API to scrape data at the game, player, and play level. We downloaded the dataset as `.csv` file from the project post on Kaggle. [1]

The data is comprised of 255 features along the column space and more than 449,371 observations along the rows. The data consists of both numeric and categorical features and describe nearly all factors that likely affect football plays. The observations span NFL games from the 2009 season through the 2016 season. A subset of the features are:

- GameID
- Drive
- Down
- Pass Yards

A. Processing Requirements

Because of the size and variety of these data, it will be important to process the data for use with our models. We expect to perform various steps to prepare the dataset for analysis and prediction. We will need to perform general cleaning of the dataset, including dealing with missing values and outliers. We will need to perform feature regularization in order to avoid numeric computation issues during the modeling process. We expect to perform encoding for the categorical variables. We also expect to to perform feature selection – while having this much data for deep learning applications is great, it is likely that some of the data is not necessary or even harmful towards the goal of prediction.

IV. Related Work

The development of predictive models in sports analytics spans various machine learning techniques, each contributing unique strengths and limitations in predicting outcomes for games like those in the National Football League (NFL). Across the studies, researchers have experimented with models like decision trees, logistic regression, and neural networks, each chosen to balance interpretability and predictive power. While traditional methods like decision trees and logistic regression are often easier to interpret, ensemble methods, particularly random forests, tend to yield higher accuracy due to their ability to generalize across diverse datasets. For example, the metric defense-adjusted value over average (DVOA), introduced in a key study, has been instrumental in enhancing predictive insights by adjusting for defensive strengths.

Neural networks have shown promise in predicting NFL game outcomes, with researchers applying deep learning frameworks seen in other sports such as soccer and horse racing. Notably, neural networks excel in accuracy, achieving rates up to 75% in specific play-type predictions, however, these models come with challenges. Their complexity often renders them impractical for real-time use by coaches, a limitation addressed by hybrid models that aim to preserve accuracy while offering quicker interpretability through simplified structures, such as decision trees that maintain 86% of a neural network's accuracy.

A notable challenge across predictive models in the NFL has been the issue of generalization. Models trained on game state information like field position, score, and remaining time tend to overgeneralize, often predicting the most common outcome by default (e.g., minimal yard gains), resulting in lower accuracy. This limitation is compounded by the lack of player-specific and team-specific data in many models. Studies indicate that introducing more granular data, such as player performance metrics, could improve model accuracy significantly. Furthermore, a common limitation lies in overfitting historical data without accounting for the dynamic and situational nature of live sports, where tactics shift based on in-game factors like injuries and tactical substitutions.

In response to these limitations, our proposal seeks to build upon these foundational insights. By integrating player-specific data and exploring methods like ReLU activations and softmax outputs within a refined neural network architecture, our model aims to achieve enhanced prediction accuracy and adaptability in NFL play-type categorization and game outcome forecasting. This approach not only addresses the interpretability challenge by emphasizing explainable components but also endeavors to mitigate overgeneralization by leveraging richer, contextual data, positioning our model to offer more actionable insights in real-time sports analytics.

V. Intended Experiments

We currently plan to perform three primary experiments, although throughout the course of the project we will likely find ourselves performing more.

A. Experiment 1: Sequence Model

We intend to develop a sequence model (`RNN`, `LSTM`, `Encoder-Decoder`, etc.) to predict the next-play vector based on previous plays in the drive. The next-play vector will consist of all of the features from the previous plays in the drive.

The goal and motivation of this will be to predict the subsequent play in a drive in order for a hypothetical coaching staff to be able to scheme and strategize against the predicted play; American football can be a complicated and multidimensional sport, and strategy has played an ever-increasing role in the modern game. By using a sequential model, we expect to capture dependencies between plays within a drive, which can offer real-time insights into the strategic choices available to an offense. Ultimately, the outcome of this experiment should provide a probabilistic estimate of the next play type and characteristics, which could enhance in-game decision-making for coaches.

B. Experiment 2: Predicting Play Distance

We intend to develop a neural network of a fully-connected architecture to predict the yardage gained or lost (play distance) on a given play. The play distance will be a continuous variable. The goal and motivation of this experiment will be to provide defenses with tools to allow them to predict the success of a particular defensive formation and scheme against a given offense. Once again, this will allow defenses to improve their play by using offensive patterns to predict what the next play might look like.

The neural network will use features such as play type, down, distance to the goal, and situational game information, which are likely to impact play distance. Regularization, feature scaling, and thorough error analysis will be important for ensuring that the model captures realistic play distances without skewing predictions. A well-performing model here will be able to provide actionable insights to defensive units about play likelihood and expected yardage, potentially influencing game outcomes.

C. Experiment 3: Run/Pass Classification

Perhaps the simplest experiment we plan to perform is classification of play type (Run/Pass) based on other features at the play level. The play type variable will take the form of a binomial variable. In the game of American football, teams can advance the ball down the field using a pass or a run. Being able to predict which play type might be seen next can allow defenses to properly assign personnel and scheme against the subsequent play.

While the model is simpler, there are still challenges to address. Run/pass imbalances, feature importance, and accuracy in high-variance situations (e.g., third down with short yardage) require careful handling. Successful outcomes from this experiment will provide insights that can improve real-time play prediction accuracy, offering teams an edge in defensive play-calling.

VI. Methods and Preliminary Experiments

We have completed our initial experiments with two of our models, namely the primary experiment: a sequence model, as well as the run/pass classification model. Much of the work in developing these models arose with organizing the data into the correct form, which we will discuss more thoroughly in the model descriptions. However, we found that both models performed well for initial experimentation, with both models outperforming random guessing for their respective outputs.

For both models, we used the following features to reduce the dimensionality of the data. For the run/pass classification model, we also performed PCA and used those loadings as features in our dataset.

- `game_id`
- `yardline_100`
- `quarter_seconds_remaining`
- `half_seconds_remaining`
- `game_seconds_remaining`
- `quarter_end`
- `drive`
- `sp`
- `qtr`
- `down`
- `goal_to_go`
- `ydstogo`
- `ydsnet`
- `yards_gained`
- `shotgun`
- `no_huddle`
- `home_timeouts_remaining`
- `defteam_timeouts_remaining`
- `defteam_score`
- `away_timeouts_remaining`
- `timeout`
- `defteam_timeouts_remaining`
- `total_home_score`
- `posteam_timeouts_remaining`
- `posteam_score`
- `total_away_score`
- `defteam_score`
- `score_differential`
- `defteam_score_post`
- `score_differential_post`
- `touchdown`
- `play_type`

A. Experiment 1: Sequence Model

We developed an LSTM model to predict the next-play vector based on a sequence of previous plays within each drive. To prepare the data for this sequence modeling, we organized the dataset by grouping each drive using a unique drive identifier, created by combining the game identifier and drive number. This approach enabled us to batch the data by game drive, capturing the continuity of plays within each drive. Overall, our dataset contained 58,729 drives, where each row vector represented a single play, and each column vector captured a specific feature related to that play. For each play, we selected 35 relevant features, such as down, distance to the goal, and yardage gained, resulting in a dataset with dimensions (58,729, number of plays per drive, 35 features per play).

Since each drive varied in length, we used zero-padding to ensure that each drive matrix had the same number of play vectors, choosing 34 plays as the standardized drive length based on the longest drive in the dataset. This choice allowed us to keep the matrix dimensions consistent across drives, making our data shape (58,729, 34, 35). This preprocessing step was essential for training the LSTM model, as it required fixed-length sequences to effectively process the sequential relationships between plays within each drive.

For the LSTM architecture, we opted for a standard configuration. The model consisted of two LSTM layers, each with 128 neurons, with an L2 regularization term to mitigate potential overfitting. We used the $tanh$ activation function within the recurrent layers to help the model capture complex dependencies between plays. The final layer was a 35-neuron dense layer, intended to output the next-play vector based on the 35 features we were predicting. For optimization, we used the Adam optimizer and mean-squared error as the loss function, aiming to minimize the error in vector-to-vector predictions across all 35 features.

During training, we tracked accuracy and F1-score as our primary evaluation metrics to assess both the model's general performance and its ability to accurately capture play characteristics. The LSTM achieved a training accuracy of 0.5474,

which suggested moderate predictive power, but the F1-score was notably lower, at just 0.058. This discrepancy between accuracy and F1-score likely indicates that mean-squared error is not an ideal loss function for this problem. Since mean-squared error does not account for differences in prediction importance across features in a vector, it may lead the model to prioritize certain features over others, resulting in suboptimal predictions for specific aspects of the play vector.

In future iterations, we plan to experiment with alternative loss functions to improve vector-to-vector predictions. Potential options include KL-Divergence, which could offer a more suitable loss landscape by measuring the difference between predicted and true probability distributions, allowing for better alignment across all 35 features. Additionally, we may consider loss functions that incorporate feature weights to account for the varying importance of features in the next-play prediction. These refinements should help enhance the model's overall accuracy and F1-score, providing more reliable predictions for each feature in the next-play vector.

B. Experiment 3: Run/Pass Classification

The run/pass classification model was a simpler model compared to the LSTM developed for play sequence prediction. We began by performing feature selection; the selected features, as previously mentioned, included essential variables that could distinguish between run and pass plays, such as down, distance to goal, time left on the clock, and field position. To further optimize the model's performance and reduce dimensionality, we applied Principal Component Analysis (PCA), retaining the top three principal components. These components captured the most variance within the dataset, and we used their loadings as input features for our classification model. This dimensionality reduction not only improved computational efficiency but also helped the model focus on key play characteristics.

For the architecture, we chose a simple fully connected neural network, following standard practices for binary classification tasks. The model began with a normalization layer, ensuring that all input values were on a comparable scale and preventing any feature from disproportionately influencing predictions. The core of the network consisted of three fully connected layers, each with 128 neurons, ReLU activation, L2 regularization to prevent overfitting, and He normal weight initialization to manage variance and promote stable gradient flow during training. The final output layer consisted of a single neuron with a sigmoid activation function, used to classify each play as either a run (0) or pass (1).

We trained this model using the Adam optimizer, a commonly used optimizer that adapts the learning rate and accelerates convergence, and binary crossentropy as the loss function, which is well-suited for binary classification problems. To evaluate model performance, we measured both accuracy and F1-score, given that accuracy is appropriate here due to the relatively balanced distribution of run and pass classes in the dataset. The model achieved a validation accuracy of 0.7278 and a validation F1-score of 0.7377, significantly outperforming random guessing and demonstrating strong initial predictive capability.

These results indicate that the model effectively differentiates between run and pass plays, benefiting from both feature selection and dimensionality reduction. However, we anticipate that further tuning of the network's architecture, such as adjusting layer sizes or incorporating dropout layers to enhance regularization, may improve performance even further.

VII. Next Steps

We would like to experiment with the following items with our models: model architecture, model hyperparameters, loss and optimization functions, and dataset preparation.

For the sequence model, we would like to experiment with various types of sequence model frameworks, such as GRU, Encoder-Decoder, and Transformer. While the LSTM model served as a good place to start, we believe that these more complex model frameworks and architectures will increase the performance metrics of the models. In addition to LSTM, we will experiment with Transformer models, which excel at capturing long-term dependencies, to see if they offer a more robust approach for play sequence prediction. These attention-based models could prove especially valuable for complex play sequences where certain events have disproportionate impacts on outcomes.

We would also like to change the model hyperparameters, namely the aforementioned hidden layer sizes, regularization, dropout, and weight initialization. Given sufficient compute resources, we may experiment with a grid-search solution to this hyperparameters. While none of our models overfit, we expect that an optimal hyperparameter configuration will increase model performance.

While Adam is a well-known and well-proven model gradient descent optimizer, we believe that introducing Nesterov momentum and perhaps gradient clipping into the optimizer might improve results. We would also like to experiment with the models learning rate, and may use a grid-search technique to tune this hyperparameter.

For this model, we would also like to explore feature engineering and its associated impact on the results. Specifically, we can create interaction terms between key features, such as score differential and time remaining, to capture nonlinear dependencies that could impact play predictions. Additionally, we could evaluate contextual game variables, such as weather conditions, field surface, and crowd noise intensity, to understand their influence on model accuracy. Time-permitting we could layer in player-specific data, like quarterback experience and running back speed, which could further refine the model's ability to account for individual variability in performance.

Finally, we would like to experiment with our dataset preparation. This would include a reevaluation of the selected features we used in this experiment, ensuring a proper train-test split for our data, data normalization and regularization, and using the `tf.data` library to optimize our batching,

prefetching, and caching of our data. We believe that, if done properly, the dataset preparation will significantly improve model performance. Additionally, in terms of the dataset, we will try to use bootstrapping to increase the diversity of training examples, particularly for rare play types or game situations, to improve generalization. Treating different game slices, such as the first and second halves, as separate data points may also capture how play style evolves as games progress, further enhancing model accuracy. These approaches should help mitigate overfitting and make the models more robust across different types of game situations.

VIII. Member Contributions

For this section we had George and Zach work together to make contributions on rewording the abstract, introduction and related works. Additionally, they collaborated to complete the data pre-processing needed to feed the data for the models in the experiments completed for this milestone. Taylor worked on using the data to create the two models for Run/Pass Classification and the sequential model. Abner assisted Taylor on the sequential model, worked on member contributions and helped general coordination for meetings and logistics. Charlie contributed to the analysis in Sections V-VII in this document, as well as final revisions.

References

[1] Horowitz, Maksim, et al. "Detailed NFL Play-by-Play Data 2009-2018." Kaggle, 22 Dec. 2018.

[2] Matt Gifford, Tuncay Bayrak, "A predictive analytics model for forecasting outcomes in the National Football League games using decision tree and logistic regression", Decision Analytics Journal, Volume 8, 2023, 100296, ISSN 2772-6622, https://doi.org/10.1016/j.dajour.2023.100296. (https://www.sciencedirect.com/science/article/pii/S2772662223001364)

[3] Joash Fernandes, Craig et al. "Predicting Plays in the National Football League". 1 Jan. 2020 : 35 – 43.

[4] Rahman, M.A. "A deep learning framework for football match prediction." SN Appl. Sci. 2, 165 (2020). https://doi.org/10.1007/s42452-019-1821-5.

[5] Guo, Xuyi. "Neural Network Models for Predicting NFL Play Outcomes." Stanford C230, Stanford University, https://cs230.stanford.edu/projects_spring_2020/reports/38964602.pdf.