Predicting Offensive Play Types in the National Football League

Peter Lee[1], Ryan Chen[2], and Vihan Lakshman[3]

Abstract— **In this paper, we apply tools from machine learning to the burgeoning field of football analytics and predict whether a team will run or pass the ball on a given play. After training four different classification algorithms on data from the 2012-2014 NFL seasons, we developed an ensemble method that combines the predictions of our two best-performing individual models and achieved a test accuracy of 75.9%, improving upon previously published results. We also explored general trends in offensive predictability and found that teams are most predictable on late downs and in the fourth quarter. Finally, we conclude with an error analysis and assess whether our models could provide value to an NFL coaching staff.**

I. INTRODUCTION

In the National Football League (NFL), prediction has long been an indispensable part of the game as teams devote numerous hours and resources towards gleaning insights into an opponent's tendencies on the field. One area of immense interest is assessing a team's propensity to run or pass the ball in a given situation. For a defense, having a sense of whether an opposing offense will run or pass informs critical decision-making about play-calling, personnel groupings to deploy, and physical positioning on the field – choices that have a substantial impact on the outcome of a game.

In this project, we take a data-driven approach to classifying offensive play types and examine whether tools from machine learning can offer value in characterizing offensive tendencies. This topic has emerged as a very active area of research in the sports analytics community in the last year, and we looked to build upon this previous work by incorporating more domain knowledge in our models. In particular, we utilize additional features describing the offensive formation on a play as well as the overall quality of the players on a given roster.

We begin by training four independent models on data from the 2012-2014 NFL seasons: logistic regression, linear discriminant analysis, random forests, and a gradient boosting machine (GBM). The input to our algorithms was a particular play, described in terms of contextual information such as the current down and yards remaining for a first down, time remaining, the current score, field position, and offensive formation, along with metrics capturing each team's season-long tendencies and strengths. Our models then output a predicted offensive play type: run or pass. After implementing these initial algorithms, we developed a mixed model that combined the predictions of the random forest and GBM and achieved results that improved upon the marks in the existing literature.

* Final project report for CS 229 at Stanford University

[1]Department of Statistics, Stanford University pejhlee at stanford.edu

[2]Department of Management Science & Engineering, Stanford University rdchen at stanford.edu

[3]Institute for Computational and Mathematical Engineering, Stanford University vihan at stanford.edu

II. RELATED WORK

While predicting offensive play types has become a subject of tremendous interest in the past year, academic research into NFL play-calling spans several decades. Much of this early work emerged from the operations research and economics community as scholars looked to determine optimal play-calling strategies through the frameworks of game theory and decision analysis. In 1978, Carter and Machol [1] initiated this line of inquiry in their seminal paper outlining optimal fourth-down strategies which was later extended to include goal-line situations by Boronico and Newbert [2] and ultimately all plays by Jordan, et al. [3]. While our work focused on predicting what play a team will run as opposed to what a team *ought* to do, the foundation established in these early papers in identifying the most critical situational statistics for play-calling decisions proved valuable to us in selecting features for our algorithms.

In 2015, interest in applying machine learning towards football analytics began to blossom as William Burton and Michael Dickey of North Carolina State University [4] addressed this question of predicting offensive play types. After training logistic regression and random forest models on NFL play-by-play data from the 2000-2014 seasons and testing on a subset of their training data, Burton and Dickey reported a prediction accuracy of 75% with a single-game high of 90%.

Ultimately, Burton and Dickey's results marked a major breakthrough in applying machine learning to football, but their decision to test on their training set is questionable since it risks overfitting and, in general, provides little information on their model's performance on unseen data.

Building off this previous work, a team from Booz Allen Hamilton [5] presented a study at the 2016 MIT Sloan Sports Analytics Conference that sought to predict both the offensive play type and the direction of the play. By developing more sophisticated features (such as a metric measuring a team's passing effectiveness in a game) and building a pipeline that combined the results from five separate learning algorithms, the Booz Allen team correctly classified a play as a run or a pass 72.7% of the time after training on play-by-play data from the 2013 NFL season and testing on the 2014 season. In addition, they achieved 57.6% accuracy in predicting the direction of the play. To date, these results constitute the state-of-the-art in the literature and the Booz Allen team's stacked model approach of combining the

results of multiple algorithms seemed particularly effective and served as the inspiration for our own mixed model.

III. DATA SETS AND FEATURES

As far as we can tell, the play-by-play data that has been used in previous work only describes the state of the game with features like down and distance, score margin, and field position. One of the ways we hoped to gain predictive accuracy is through better input data. Specifically, we sought data describing the personnel on the field for each play and measures of their quality. We acquired this raw data from two sources: the NFL statistics and analytics blog Football Outsiders (`www.footballoutsiders.com`) and a repository of historic Madden NFL video game ratings for each player hosted at `maddenratings.weebly.com`.

A. Football Outsiders Play-by-Play Data

Football Outsiders tracked information about personnel groupings (i.e. the numbers of wide receivers, running backs, defensive linemen, etc.) on the field for each play from the 2011-12 season to the 2014-2015 season for use in their own analyses. However, these data are proprietary and we are grateful to Aaron Schatz, editor in chief of Football Outsiders, for granting us access for this project. In addition to data detailing the numbers of players at each position, their data set also includes auxiliary information such as players lining up in atypical positions, whether or not the offense was in shotgun formation, whether or not a quarterback run was a designed draw or a coverage-forced scramble, etc. Of all the various features included in the Football Outsiders data set, we chose to use the following:

1) Score Difference
2) Current Quarter
3) Time Left in Quarter
4) Current Down
5) Distance to First Down
6) Count of Offensive Players per Position
7) Count of Defensive Players per Position
8) Formation (e.g. shotgun, no huddle)
9) Indicator of an offensive player out of position
10) Turnovers
11) Indicator of whether offense is at home

B. Madden Ratings

We also sought to incorporate more domain knowledge into our models by taking into account the strengths of a given team. For example, a squad with outstanding running backs or an offense facing a team excellent at defending passes are both more likely to run the ball. At `maddenratings.weebly.com`, each iteration of the video game (released once per year) has 32 files, one for each NFL team. Within each team-season file, each row describes one player – name, position, jersey number, overall rating, and each of the component attribute ratings (some weighted combination of which precisely defines the player's overall rating). For our purposes, the player's overall rating was all that we needed to capture since the purpose of using these data was to provide some generic measure of player quality. In fact, including component attribute ratings (e.g. speed, agility, strength, throw accuracy, awareness) may increase the risk of overfitting.

Using the Madden player ratings and snap count data from Football Outsiders, we derived scores for seven position groups on each team: quarterbacks, running backs, receivers, offensive line, defensive front, and secondary. To compute these scores, we weighted each player's Madden rating by the percentage of snaps he played that season and then summed these weighted scores for players in the same position group. Our reasoning behind weighting by the percentage of snaps played was to avoid penalizing teams for having low-rated backup players who never see the field.

C. Derived Features

In addition to the position group scores computed via Madden ratings, we calculated additional derived features to provide information on a team's season-long and in-game tendencies to complete our full feature set.

13) Proportion of pass plays called over week, season, last 50 plays
14) Proportion of pass plays faced by defense over week, season, last 50 plays
15) Indicator of score difference greater than 7
16) Quarterback pass completion rate over week, season, last 25 plays
17) Weighted Madden rating of each offensive/defensive position group.

IV. TECHNICAL APPROACH

In this section, we describe the various learning algorithms we trained and provide an overview of the relevant performance metrics we considered to analyze our results. As a matter of notation, we will denote a single training example, corresponding to a play, by $x^{(i)}$ with a corresponding label $y^{(i)} \in \{0,1\}$ for $i \in \{1,2,\ldots m\}$ where m denotes the number of training examples. In our representation, we let $y^{(i)} = 0$ denote a run and $y^{(i)} = 1$ a pass and let $\hat{y}^{(i)}$ denote the label predicted by our algorithm for the ith training example. To implement these algorithms, we utilized Python's `scikit-learn` library [6].

A. Preliminary Models

1) Logistic Regression - A natural algorithm for our binary classification problem is logistic regression which produces outputs in the range $[0,1]$ via a hypothesis that takes the form of the logistic function

$$h_\theta(x^{(i)}) = \frac{1}{1+e^{-\theta^T x^{(i)}}}$$

The parameter θ is fit by performing gradient descent on the following loss function where we add an ℓ^1 regularization term.

$$\ell(\theta) = \sum_{i=1}^{m} -(y^{(i)} h_\theta(x^{(i)})) + (1-y^{(i)})(1-h_\theta(x^{(i)})) + \lambda ||\theta||_1$$

If the output of the logistic function is greater than our classification threshold of 0.5, we set $\hat{y}^{(i)} = 1$ and, conversely, for any output less than 0.5 we set $\hat{y}^{(i)} = 0$

2) Linear Discriminant Analysis (LDA) - As an alternative to logistic regression, we also considered a generative model which attempts to learn the defining characteristics of a run play and a pass play and then classifies a new example by determining whether it is more similar to the former or latter category. The LDA algorithm is a generative model that projects the input data onto a two-dimensional subspace and then fits a linear boundary between the two learned classes to make a prediction. As a result, LDA is also commonly utilized as a dimensionality reduction technique. LDA assumes that its features are drawn from a multivariate Gaussian distribution with a mean vector and common covariance matrix across each class. It uses a Bayes classifier to assign an observation $x^{(i)}$ to the class k in which the probability

$$\delta_k(x^{(i)}) = x^{(i)^T}\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\sigma^{-1}\mu_k + \log\pi_k$$

is highest. Here Σ denotes the covariance matrix, μ_k the mean vector, π_k the proportion of training examples that belong to class k, and $\delta_k(x^{(i)})$ the probability of training example $x^{(i)}$ belonging to class k.

3) Gradient Boosting Machine (GBM) - Taking on a different paradigm for approaching this classification problem, boosting is an ensemble technique that combines the performance of several weak-learning algorithms that perform slightly better than random into a very strong classifier. In the `scikit-learn` implementation of gradient boosting, each weak learner takes the form of a decision tree. The algorithm performs gradient descent on a differentiable loss function to assign weights to training examples misclassified by previous weak learners and then generates a new decision tree that does well on precisely these missed examples. After tuning, we set a hyperparameter of 300 weak-learning decision trees in our model.

4) Random Forest - One of our more effective methods was a random forest, which, like the GBM, is an ensemble of decision trees that produces a better prediction when combined together. Decision trees, while very easy to understand and simple to use, are not very good at making any type of prediction on their own. One of the first ways to reduce a decision tree's high variance is to apply bagging (bootstrap aggregating). After bootstrapping B separate training sets, we would average the results of these bootstrapped training sets to create a single, lower-variance statistical model

$$\hat{f}_{\text{bag}}(x^{(i)}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{\star b}(x^{(i)})$$

As with the GBM, we found that 300 trees gave us the best results.

B. Mixed Model Approach

After training our four initial algorithms, we then examined whether we could combine the predictions from these models to develop a stronger classifier. After experimenting with assigning weights to the output of each algorithm, we found that taking a weighted average of the outputs of the GBM and random forest gave us the best results with a 60% weighting given to the GBM and and 40% weighting to the random forest. Then, we again make a prediction by setting $\hat{y}^{(i)} = 1$ if the output of the mixed model algorithm is greater than the classification threshold of 0.5 and 0 otherwise.

C. Error Analysis Metrics

For classification problems, a very intuitive metric for assessing the performance of an algorithm is *accuracy* which measures the proportion of examples classified correctly. Formally, accuracy is given by $\frac{1}{m}\sum_{i=1}^{m}\mathbb{1}\{\hat{y}^{(i)} \neq y^{(i)}\}$

In the context of NFL play-calling, predicting incorrectly can have very adverse consequences as defensive formations designed to stop a run may be very ill-equipped to defend a pass and vice versa, and we wanted to see if our model was more likely to make a certain type of misclassification. As a result, we considered two additional metrics *precision* and *recall* that assess such performance. The former metric measures, for each class, the proportion of true positive classifications while the latter measures, for a given class, the ratio of true positive labels to the total number of examples assigned that predicted label. Formally, for a classification problem with k classes, we can define the precision p_k and recall r_k of the kth class as

$$p_k = \frac{\sum_{i=1}^{m}\mathbb{1}\{\hat{y}^{(i)} = k \wedge y^{(i)} = k\}}{\sum_{i=1}^{m}\mathbb{1}\{\hat{y}^{(i)} = k\}}$$

$$r_r = \frac{\sum_{i=1}^{m}\mathbb{1}\{\hat{y}^{(i)} = k \wedge y^{(i)} = k\}}{\sum_{i=1}^{m}\mathbb{1}y^{(i)} = k\}}$$

where in our binary classification problem $k \in \{0,1\}$

Finally, we also plotted a receiver operating characteristic (ROC) curve which plots the true positive rate vs. the false positive rate as the classification threshold varies. If our classifier is strong, then the true positive rate will increase very quickly. With the ROC curve, we can also examine the area under the curve, known as the AUC, which can be interpreted as the probability that a random example with a true label of 1 will be assigned a higher score by the algorithm than a random example with a label of 0. Thus, an AUC score close to 1 indicates that a classifier is performing well.

V. Results & Discussion

A. Overall Accuracy

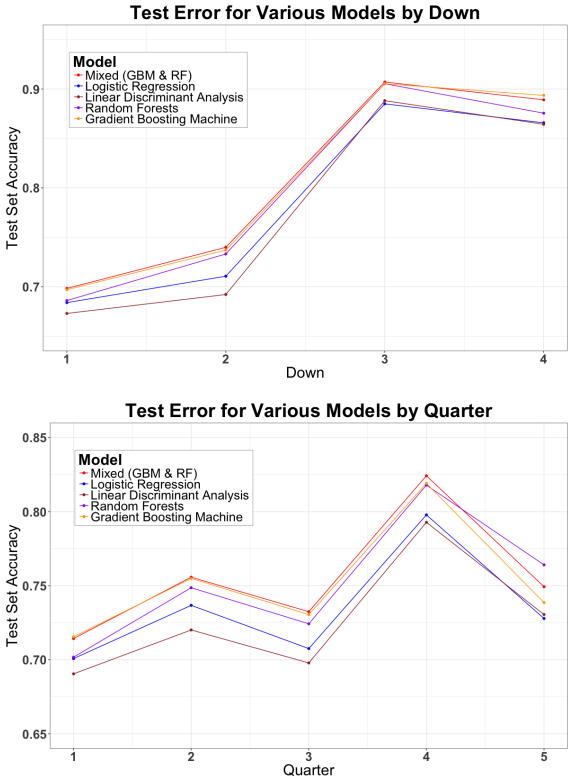
After training our algorithms, we looked to measure how well our model would generalize to unseen data by implementing 10-fold cross validation and then testing on 10% of the original data, which we did not train on, to report final results.

Model	Training Accuracy	Test Accuracy
Logistic Regression	0.738	0.737
Linear Discriminant Analysis	0.731	0.727
Random Forest	0.758	0.751
Gradient Boosting Machine	0.764	0.757
Mixed	**0.763**	**0.759**

As mentioned earlier, the mixed model, obtained by taking a weighted average of the outputs of the GBM and random forest achieved the best results with an accuracy of 75.9%, an improvement over the benchmarks in the literature. Moreover, we see that, across all models, our training accuracy was marginally higher than our test accuracy, suggesting that we were not overfitting our data by a substantial amount.

B. Situational Results

On top of examining the overall accuracy of our models, we also considered the performance of our algorithms on a down-by-down and quarter-by-quarter basis (where the fifth quarter denotes overtime).



In both plots, we see very little deviation in the performance of our algorithms relative to each other, suggesting that our best models tended to outperform the others across different inputs. We also observe that our accuracy increases on later downs, which likely stems from the fact that teams become more concerned with gaining enough yards to move the chains on third and fourth down which limits the scope of available plays. Similarly, we find that teams become more predictable in the second and fourth quarters, most likely because the ends of these quarters have outsize effects on the outcome of the game. In particular, 4th quarter accuracy is highest because score margin and time remaining often directly dictate play-calling in end-of-game scenarios.

Conversely, we found that many of our misclassifications came in the first and third quarters where teams often deviate from their tendencies by making adjustments both before a game and at halftime.

C. Assessing Team Predictability

As a consequence of our results, we can assess which NFL teams tended to be the most predictable over the course of a game and a season as well as which teams emerged as unpredictable play-callers, a metric that also provided further insight into our error analysis.

Year	Week	Team	Accuracy
2013	11	Tennessee	0.946
2013	8	Tampa Bay	0.943
2014	7	Dallas	0.934
⋮	⋮	⋮	⋮
2012	12	Philadelphia	0.551
2013	14	San Francisco	0.522
2013	6	Denver	0.472

TABLE I
BEST AND WORST PREDICTED GAMES

Year	Offense	Accuracy
2014	Dallas	0.860
2014	San Diego	0.830
2012	Arizona	0.823
⋮	⋮	⋮
2014	Seattle	0.685
2013	Seattle	0.680
2014	Miami	0.676

TABLE II
BEST AND WORST PREDICTED TEAM-SEASONS

From examining the teams on whom our algorithm accumulated a large number of mistakes, we noticed that many of those squads featured mobile quarterbacks, headlined by Seattle (Russell Wilson), Carolina (Cam Newton), and Philadelphia in 2013 (Michael Vick). Upon further examination, we found that we were inaccurately classifying quarterback scrambles 77% of the time since such plays are designed to be passes but end with the quarterback running. Fortunately, the Football Outsiders dataset indicated whether a quarterback run was a designed call or a scramble stemming from an abandoned pass play. After relabeling all such scrambles as pass plays, we were able to improve our accuracy from our initial results presented in our poster by over 1%. From our current results, we still see that teams with mobile quarterbacks are difficult to classify for our algorithm which, we hypothesize, is due to the fact that dual-threat signal-callers provide more flexibility and hence an additional layer of unpredictability for an offense.

D. Error Analysis

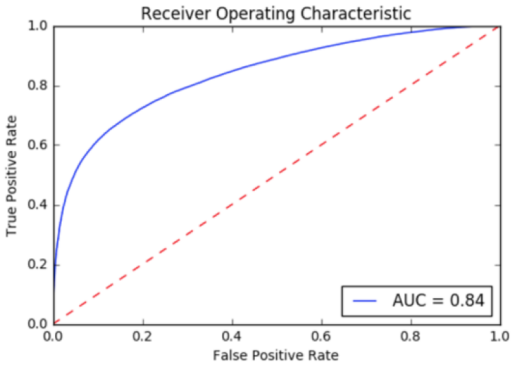
Below we include the *confusion matrix* for our classifier, which charts the total number of plays we classified correctly as well as the number of run plays we labeled a pass and vice versa.

		Predicted	
		Run	Pass
Actual Class	Run	28174	10598
	Pass	13937	49044

As we discussed earlier, these metrics are very important in the application of play-calling. From the confusion matrix, we see that we mislabel a pass play as a run 22.1% of the time and mislabel a run as a pass approximately 27% of the time. While, as a matter of future work, we aim to improve both of these errors rates, this behavior of misclassifying a greater proportion of run plays is much preferred to the alternative since pass plays, on average, result in more yards and can thus be more damaging to a defense, particularly for an unsuspecting one.

From the confusion matrix, we can also calculate the precision and recall scores across each label:

Play Type	Precision	Test Recall
Run	0.670	0.727
Pass	0.822	0.779



We also plot our ROC curve above with the corresponding AUC value, suggesting that our classifier would assign a random pass play a score higher than a random running play with probability 0.84, a sign that our classifier generalizes well.

VI. Conclusion and Future Work

In this project, we apply machine learning techniques to predict offensive play types in the NFL, using a rich dataset that provides detailed information on player personnel and formations and augmenting this data with scores to measure player quality on a given team. After training four classification algorithms, we found that the ensemble tree methods of random forest and gradient boosting outperformed both logistic regression and linear discriminant analysis most likely because the latter two models attempt to find a linear decision boundary which does not need seem to lend itself well to this particular problem. We were then able to combine our two top-performing methods into a mixed model that achieved a test accuracy of 75.9% which marks an improvement over previously published results.

In the future, we would like to extend our work in several exciting ways. One particularly intriguing avenue is extending our models to predict not only the type of play but also the direction (either left or right). While being able to effectively predict run vs. pass plays is certainly valuable for any NFL defensive coordinator, further information on the direction of the play could be even more valuable in gaining a competitive edge. As an extension of this idea, we would also like to look into making our algorithms more granular and predict specific play types such as a "screen" or "draw" play for runs and route combinations on pass plays. Football Outsiders has already done some work in labeling plays with this level of granularity and, as this data becomes more robust and available, we would be interested in applying our models towards predicting more specific play types.

Moreover, previous studies predicting NFL play-calling all utilized timeouts remaining as a key feature in their models, but we were unable to do so because our dataset did not specify which team called the timeout. With additional time, we could look into augmenting our dataset with this information, which we suspect would improve our results. Another important feature that we did not have in our data that would almost certainly improve our results is the weather, and we would also like to incorporate that information into our models.

Finally, we would also be interested in looking into the possibility of building a web or mobile application that NFL coaches could use to input in-game situational statistics and obtain a play prediction. Currently, the NFL does not allow outside technology on the sidelines or in the coach's booth outside of still pictures, but with many expecting the rules to change in the near future, tools that can contribute to play-calling decisions could emerge as extremely valuable and our project, with this additional work in the future, might be able to provide such value.

VII. Acknowledgments

We would like to thank Prof. Ng and Prof. Duchi for leading an excellent course and to the CS 229 TAs for their helpful feedback and comments on this project. We are also especially grateful to Aaron Schatz and Football Outsiders for providing us with their proprietary play-by-play data.

References

[1] V. Carter and R. E. Machol. Optimal strategies on fourth down. Management Science, 24(16):1758 1762, 1978.

[2] J. Boronico and S. Newbert. An empirically driven mathematically modeling analysis for play calling strategy in American football. European Sport Management Quarterly, 1:2138, 2001.

[3] J. D. Jordan, S. H. Melouk, and M. B. Perry. Optimizing football game play calling. Journal of Quantitative Analysis in Sports, 5(2):134, May 2009.

[4] W. Burton and M. Dickey. "NFL play predictions". In: *JSM Proceedings, Statistical Computing Section*, 2015.

[5] Booz Allen Hamilton. Assessing the Predictability of an NFL Offense. MIT Sloan Sports Analytics Conference, 2016.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:28252830, 2011.