Predicting NFL Play Call on Third Down

Ziv Schwartz (zs1349), Esteban Navarro Garaiz (eng272)
Charles Brillo-Sonnino (cbs488), Santiago Novoa (smn405)
New York University

December 19, 2018

1 Abstract

For the final project, our group is interested in trying to predict the type of play that will be called on third down in a given NFL game. Third down conversions are one of the key metrics that can measure a team's success and win probability. On the other hand, stopping an opponent on third down is critical for the defense. We utilized a massive dataset [1] from Kaggle containing every event (play or otherwise) for every NFL game between the years 2009 and 2017. The dataset contains a total of 102 features, ranging from date and score to touchdown probability and win probability. After performing feature selection, we implemented different models and found that a Random Forest model captures the nuances in the data the best.

2 Introduction and Motivation

In American Football, governed by the National Football League (NFL), the goal of the team on offense is to score by moving the football to the other end of the field. They can do this by either running with the ball or passing the ball. The team has 4 attempts, referred to as "downs", to move the ball at least ten yards down the field. If they succeed, they get 4 more attempts to keep moving down the field. If they fail, the ball is turned over to their opponent. Generally, teams use fourth down to punt (kick) the ball down the field so their opponent has to cover more distance in order to score. This makes third down critical in determining whether the offensive team's drive can continue. If a team is able to gain the necessary yardage on third down, this is known as a "third down conversion".

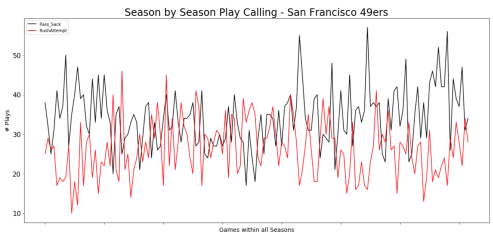

Figure 1: Number of passing and running plays called by the San Francisco 49ers during each game in our sample

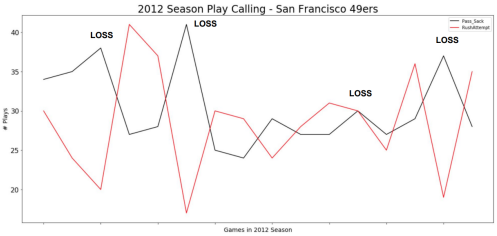

Figure 2: Number of passing and running plays called by the San Francisco 49ers during each game of the 2012 season

The goal of this project is to predict whether a team will run or pass the ball on third down. The result of this prediction has many applications for deployment but it is of particular importance for the defense. Knowing what type of play is about to be run is extremely valuable information, as their target is to stop the offensive team from getting a third down conversion. The most difficult part of this problem is that in the NFL, there are many factors that can influence a coaches play-calling decisions. For example, teams facing long third downs tend to pass the ball more. However, if the team with the ball has a lead and it is towards the end of the game, their goal might become exhausting time instead of converting the third

down, thus choosing to run. How a coach makes that decision can vary wildly across teams and is subject to human errors. In addition, coaches on offense may actively try to be unpredictable in their play calling to gain the upper hand over the defense. Our hypothesis is that these decisions, depending on the game scenario, are likely to have patterns. Therefore, making a reasonable prediction about the type of play to be executed next is possible and can give the defense a significant competitive advantage. Figure 1 shows the number of running and passing plays called by the San Francisco 49ers in every game in the dataset. As the graph shows, the proportion called per game is hardly stable. Similarly, graph 2 shows the number of running and passing plays called by the team during the 2012 season. Overlaying are the results for the games. We can see a clear trend: the team passed much more in the games they lost - presumably because they are trying to catch up in the score - and ran more in all games they won. This hints, as hypothesized, to the data having underlying patterns that can be picked up by our modelling.

3 Methodology

For our analysis, we utilized the Detailed NFL Play-by-Play Data 2009-2017 from Kaggle that contains records for all events in a game, from all games in the NFL between the years 2009 and 2017. It contains 407,688 instances with 102 features. To get a better understanding of the data, Figure 4 shows how likely a team is to call a run or pass by the different downs. Passes are much more prevalent on third down, when the team needs to convert to keep the drive alive. We do not consider fourth down since the dominating majority of the plays called are punts, with a very small percentage being a run or pass. Figure 3 shows the distribution of plays per quarter.

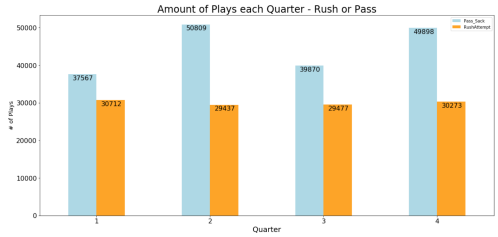

Figure 3: Plays per Quarter

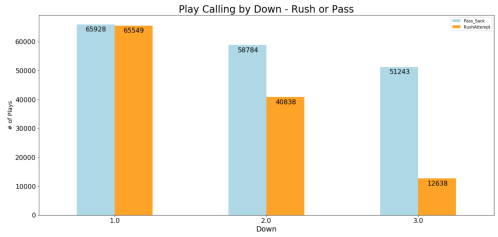

Figure 4: Plays per Down

For our purposes, we combined Pass Attempts and Sacks, as both are initially called as a passing play (a sack is a pass where the quarterback gets tackled before he can throw the ball). Our problem focuses on classifying which play will be called on third down.

The data's features range from date, game ID, and teams playing to more unique features like touchdown probability, yards after catch, and run location. Given this large feature size (102), it is not smart nor computationally efficient to use all these features. We narrow down the feature list to a set that reflects our prior domain knowledge regarding what will be the most important. The entire feature set after feature selection can be found in the Appendix. In addition to the given features, we built a few new features. First, we created a variable for each individual game that measures the moving average of net yards from both passing and running plays for each team's offense. These features can help indicate what type of play has been succeeding in the current game, given the particular circumstances and can be used as an indicator of what will be called next. Along with this, we created a binary indicator for a "two-minute drill": a situation when the time remaining in a half is under two minutes – important as teams are trying to score before time runs out and they lose possession of the ball, or the game ends.

We split our data into training and testing sets, using an 80%/20% split. We implemented five different classification models to compare performance, as all of these models have advantages and disadvantages for different classification problems. The five models studied were: Decision Trees, Random Forest, Logistic Regression, K-Nearest Neighbors, and Gaussian Naive Bayes. All models were set to the default scikit-learn parameters, except for Logistic Regression which we passed with L1 and L2 regularization. To evaluate the models we utilized the Area Under the ROC (AUC) curve metric and after establishing which model performed best, we dove deeper into the model's hyperparameters to maximize the AUC.

4 Results

Figure 5 shows the AUC for the out-of-the-box models tried. We can see that Random Forest is clearly the best model and, given the data, this is not surprising.

First, Random Forests can handle all sorts of data without the need to scale or change their types. Our features are binary (2Min, GoalToGo), probabilities (TD, FG, Safety) and numerical (Yrdline100, TimeSecs). This means that Random Forest is a natural candidate.

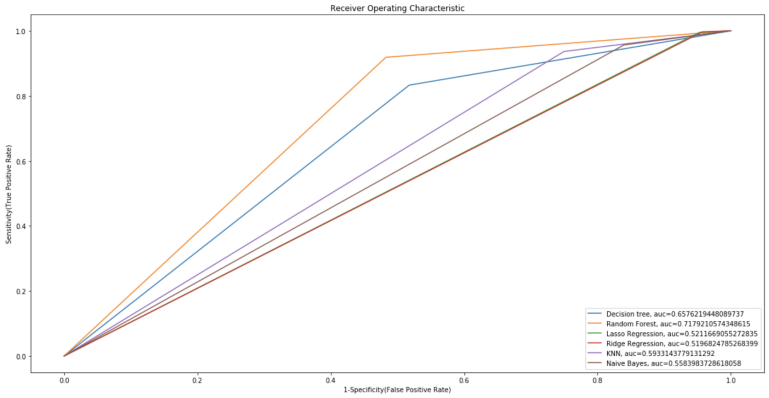

Figure 5: ROC curve for all six models

Second, Random Forests are good when trying to learn a specific data set. They have high variance, as they are very sensitive to sampling error, but come with low bias, as they can learn the given data as you let the forest grow. As we have discussed before, NFL decision making is nuanced and scenario dependent: teams who are leading tend to run late in games, teams losing tend to pass more, teams close to their own goal line run to minimize the risk of a safety, etc. This means multiple decision trees working as an ensemble are a good prediction candidate.

Random Forests are as close as it comes to a free lunch in Data Science: they decrease the variance of individual trees, without increasing bias, by using bootstrap aggregation. This selects multiple random samples with replacement from the training set, fits a tree to each one and predicts on the unseen examples by a majority vote classification. If the trees are uncorrelated, and that is the sole goal of bootstrap aggregating or "Bagging", this decreases the variance without augmenting the bias. Furthermore, Random Forests take the original bagging algorithm for trees and modifies it slightly to decrease feature correlation as well. In each split, the learning process is only allowed to use a random subset of the features. Typically, if there are n features available, the algorithm utilizes $\sqrt{n}$ for each split.

Another big advantage of Random Forests is that they naturally allow us to rank feature importance by computing out-of-sample error before and after permuting a feature. Figure 6 shows the feature importance for the features used in the model. While the Two Minute Warning binary variable does not make a big difference, both running and passing moving averages, add information to the model. This, intuitively, makes sense: if a team is being particularly successful in one type of play, it will tend to keep choosing that play.

5 Conclusion and Discussion

The results show that the Random Forest model is in fact capable of picking up some of the nuances in the data and predict it successfully. As we can see in table 1, the confusion matrix is concentrated mostly along the diagonal, as the model has a 84.21% accuracy, a 87.83% precision and a 92.21% recall. By further exploring the available features, it is possible the model could be improved. However, the fact that we are achieving these metrics on an unbalanced class after reducing the features by 90%, speaks to the power of using Random Forests for this particular problem.

One limitation of this model is that we have only considered run, pass and sack as play type in this first iteration. The dataset records every play that happened in a game, even including those plays that were called as penalties. In the NFL, the results of plays can be nullified after it has already occurred because of

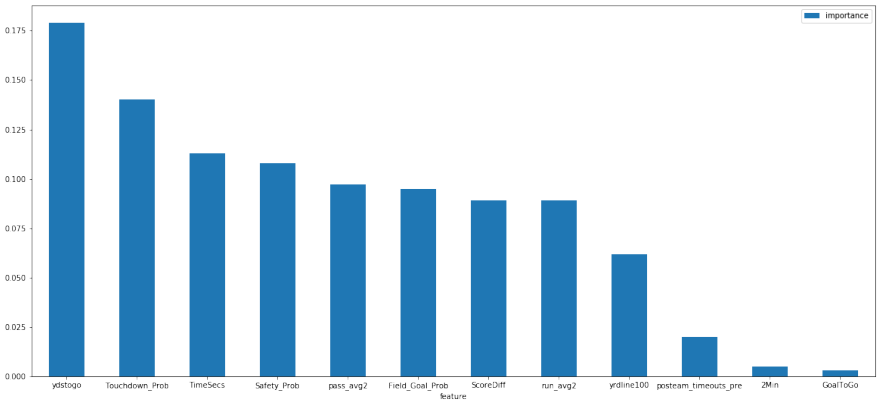

Figure 6: Feature Importance for the Random Forest model.

True / Predicted (raw and %)	Predicted run	Predicted Pass
Run	1314 (10.79%)	1238 (10.17%)
Pass	684 (5.61%)	8937 (73.41%)

Table 1: Confusion Matrix for our best model: Random Forest

a penalty but the down often remains the same. It would be advantageous to include these plays when they occurred on third down to the model. However, it is often difficult to determine the original play type due to the multitude of different penalties in the NFL. In addition, it would be interesting to consider the effect of a penalty on the subsequent play call. For example, if a team passes the ball on third down but it gets called for a penalty, would they now be more or less likely to try passing the ball again.

6 Future Work

Our project is only a start to the type of analysis and predictive modelling that can be done for play type. In the future, our team would like to focus on optimizing the feature selection. Our initial feature selection stemmed mostly from our intuition about American Football and the way the NFL operates. While this is a good first approach, we can further analyze the statistical significance for every feature when predicting play type. It will also be important to consider the correlation between the different features in order to make sure the model does not suffer from multicollinearity - where two or more features are heavily correlated. These statistical methods, coupled with our initial intuitive approach, may lead to better feature selection and improved model performance.

Although there are many features present in this dataset, it is possible that the addition of external features could give a significant boost to model performance. For example, one could bring in outside data on the prowess of specific quarterbacks, running backs, and defenses. This could improve the analysis by tuning the model to be conditional on specific players and teams, as our current model generalizes for all players and teams. In addition, expanding the feature engineering is another factor that could contribute to the model's success, as shown by the importance of the features created for this iteration.

Lastly, we could generalize this problem and work on predicting plays for every down, not just third down. In theory, the different trees of a Random Forest should be able to pick apart the nuances as each if we added the down number as a model feature.

A Feature List

- TimeSecs - Time remaining in game in seconds
- Yrdline100 - Distance to opponent's end-zone, ranges from 1-99
- Ydstogo - Yards to go for a first down
- GoalToGo - Binary. Goal down situation (1), else (0)
- ScoreDiff - The difference in score between the offensive and defensive teams (offensive.score - def.score). Shows if offensive team is ahead or behind.
- Posteam_timeouts_pre - Timeouts remaining for offensive team at the start of the play
- Touchdown_Prob - Probability of the possession team scoring a touchdown next
- Field_Goal_Prob - Probability of the possession team scoring a field goal next
- Safety_Prob - Probability of the possession team allowing a safety next
- 2Min - Binary for 2 or less minutes remaining in half (1), else (0)
- Run_avg2 - Moving average of net yards gained from running plays for team per game
- Pass_avg2 - Moving average of net yards gained from passing plays for team per game

References

[1] Max Horowitz. Detailed nfl play-by-play data 2009-2017. *Kaggle*, 2018.