Tackling the NFL: Quantifying Defensive Metrics

Krishanu Datta and Joe Kajon

MIT Sloan School of Management

15.095: Machine Learning Under a Modern Optimization Lens

Dr. Dimitris Bertsimas

12 December 2023

Introduction

In the NFL, there exists untapped potential for a comprehensive assessment of a defensive player's true tackling impact beyond just the absolute number of tackles in a season. Higher value should be placed on those players who consistently defy the odds by making tackles when the probabilities might suggest otherwise. Our project aims to tackle this challenge by predicting the likelihood of a defensive player executing a successful tackle at a given moment in a game and then identifying which players consistently defy the odds. By leveraging an expansive dataset that includes the player-specific and game-specific factors, we were able to achieve 94% accuracy and 88% AUC score utilizing an optimal classification tree model. In addressing this previously overlooked area, coaches can use these metrics to enhance their defensive rosters, identify player strengths and weaknesses, and uncover hidden gems in the player pool, ultimately transforming how drafts and trades are approached.

Data

The data for our project consists of various data from Weeks 1-9 of the 2022 NFL season. The data was provided and created by NFL Next Gen Stats and Pro Football Focus, accumulated for an NFL Competition on Kaggle.

The first set of data contained information regarding each **game** that was played in the first half of the season. This included data such as week, gameDate and gameTimeEastern which indicated when the game was played, as well as homeTeamAbbr, visitorTeamAbbr, homeFinalScore, and visitorFinalScore, which signaled which teams played each other and the resulting outcome. This set of data was crucial in piecing together some of the subsequent datasets.

The next dataset held information regarding each **play** of each game, headlined by variables relating to what happened on the play such as ballCarrierName, playDescription, playResult, and playNullifiedByPenalty. Moreover, the dataset also contained data about the circumstances of when the play occurred including variables such as quarter, down, yardsToGo, possessionTeam, defensiveTeam, gameClock, preSnapHomeScore and preSnapVisitorScore. Finally, general football characteristics of the play were also noted in the data, of which offenseFormation, defendersInTheBox (number of defenders in close proximity to line-of-scrimmage) and passProbability (probability of next play being pass based on an external model) stood out. This dataset provided much needed context for each play, proving especially useful in both aspects of feature engineering and evaluation of our model.

Another dataset provided was one relating to data on each **player** in the NFL. This file contained the usual player-specific details associated with athletics including height, weight, birthDate, collegeName and position. While being one of the smaller datasets we

leveraged, several of its features directly and indirectly played a role in our model's predictive capabilities.

The next dataset stored information on the **tackles** which have been made on every play up to that point in the season. The dataset had binary features of possible outcomes for a defensive player on a given play like forcedFumble, assist, missedTackle and most importantly, tackle, which indicates whether or not a player made a tackle on a given play. This variable was the target variable of our model.

The final dataset employed included **tracking** data of players at every point of every play. Not only was this data the most important to our final product, but it was the most comprehensive. The dataset included general information about the player such as displayName, jerseyNumber and club, but vitally included metrics about the player's movement including their x position, y position, speed, acceleration, distance traveled from last frame, orientation, and direction of motion. These factors aided in creating a numerical representation of how each play occurred and proved especially impactful in calculating tackle probability.

Using identifiers provided in each dataset (gameID, playerID, playID), we were able to merge all the sets of data into one large dataset in which each row contained information of important frames of each play for each game, for every player on the field. From there, we applied data preprocessing techniques and feature engineering methods to arrive at our final batch of feature and target variables.

Methods

While the data provided by the NFL was mostly clean and in good shape, there was still work to do in terms of getting the data ready to be fed into the model.

Preprocessing - Formatting Features

We started preprocessing by looking at variables such as birthDate and height that were presented in a string format but were capable of being converted to a numerical format. The birthDate variable was first converted to a datetime format which was then subtracted from the current date as a datetime object to achieve the player's age in days. Finally, by dividing by 365.25, we arrived at a numerical value for the player's age based on their given birthDate. Because there were only 16 uniquely reported heights among NFL players, we converted the heights, which were given as a combination of feet and inches, to total inches via a mapping function. To reduce the size of our feature set following one-encoding, we also decided to map player positions to commonly associated position groups. For instance, players who played cornerback and safety were both assigned to defensive back, as the positions are commonly grouped into that more general position category based on their role on the field. Additionally, we created a new

column called “timeRemaining” which indicated the total time left in a game based on the values of the quarter and gameClock variables. In this fashion, we added a dimension which measured the absolute time left in a game, rather than just the amount of clock left in a quarter. One last bit of feature formatting included aligning the given variables of home team score and visitor team score to the players on each play, namely assigning the home team score to a new variable titled “offenseScore” if the player’s team was on offense, and to “defenseScore” if the converse was true.

Preprocessing - Narrowing Project Scope

Due to the size and scale of our data, we opted to narrow the scope of our objective in a couple of different ways. Firstly, we decided to drop all plays in which penalties negated the play as this would mean the play’s outcome was invalidated by a foul. While using plays with penalties would give us additional samples, the quality of that data would be questionable since the results of the play did not actually stand in the context of their respective games. Moreover, we chose to drop plays across both our training and test set in which the plays did not end in a tackle, as we wanted our model to feel confident in predicting one player to make a tackle on each given play. Even though this means we currently drop plays which end in forced fumbles, out of bounds runs, and touchdowns, these remain outcomes we hope to incorporate in the future. Lastly, we decided that for this version of the model, we would predict tackle probabilities at the point of a reception on passing plays. Furthermore, we would focus our analysis on the frames at which a pass was caught and look to find probabilities for each defensive player at that moment. This choice simultaneously made our goal clearer and the problem more manageable. Again, while we moved towards limiting the scope of our project, this was in an effort to focus on a more accurate and refined tackle probability in the short term, with aspirations of adding back additional dimensionality to the initial model build later on.

Feature Engineering

Though the dataset we worked with was extensive in its list of potential variables, we identified a slew of quantifiable opportunities which could help the model predict at a higher rate. We sought to address these areas by formulating, coding and appending new variables to the dataset based upon the given slate of variables.

One of the first thoughts we had was quantifying the tackling abilities of a defensive player and packaging that quantity as an input into the model. We chose to accomplish this idea by adding new columns for cumulative tackles (and missed tackles) by a player over the course of the game they are playing and over the course of the season as a whole. This way we could capture the players’ rolling tackle count for the season as a proxy for their overall tackling ability, and the tackle count in the given game as a representation of the players’ “hot hand” (or lack thereof) if they’re having a particularly

good game. We were able to create these columns using the Python function "cumsum()" which can return the cumulative sum of a series object.

Next, we invented our signature "distance to ball carrier" variable, which calculated the distance from the defensive player to the offensive player who made the reception at the time of catch. This feature was formed using a simple Euclidean distance calculation (square root of sum of delta x's and delta y's), but its impact was anything but insignificant. We abstracted the results of this variable determination to generate the related variable "rank of distance to ball carrier", which ranked the defensive players by their distance to the ball carrier. As one can imagine, these two variables proved to be two of the most impactful in our final models as how close a defender is to the player they are trying to tackle as well as how close they are relative to other defenders would be an intuitive factor in the likelihood of that defender being the one to make the tackle.

Continuing with this idea of leveraging the path from the defender to the offensive player, we proceeded to produce a new variable which tallied the number of offense players in between the player and ball carrier in order to get a value for the number of potential blockers in the way of the potential tackler. We obtained this metric by grouping the number of offense players on each play which had a smaller distance to the ball carrier than the defensive player and summed over this grouping. In a similar vein, we then took this group of offensive players which stood in between the tackler and their prey and totaled their combined weight to implement yet another new variable which measured the weight of offense players in between player and ball carrier. The methodology behind the creation of this variable was that heavier offensive players are not only harder "obstacles" to get around, but they are also more likely to be linemen whose entire job is to block defensive players, making the defensive players' task inextricably inversely linked to the mass of the offensive players in their way.

From here, we decided to improve on the mechanics information given by the dataset. The data offers the speed and acceleration of each player in the direction their body is currently moving. However useful this information is on its own, we realized that it would be more useful to have data on the speed and acceleration of each defensive player towards the ball carrier. Utilizing fundamental physics calculations, we were able to derive the angle of each player to the ball carrier and subsequently calculate the resultant velocity and acceleration vectors in the direction of the ball carrier. These variables offered immense promise as players whose momentum was carrying them towards the player who made the catch should have a better shot at making a tackle than players who were going faster or accelerating at higher rate but in the opposite direction, and this metric was able to home in on that fact.

All in all, our feature engineering phase capitalized on the already rich dataset and paved the way for high quality modeling due to high quality data.

Modeling

Before fitting our data to any model, we checked the columns for pairwise multicollinearity using a correlation matrix and VIF scores. This analysis revealed certain metrics which were highly correlated due to the nature of their data (for instance delta x and delta y were correlated with distance) and we promptly removed the appropriate columns to ensure this was not the case going into model training. In addition to removing features which had overlapping information, we also dropped features which were not significant upon initial tests, weeding off unnecessary variables and reducing the chances of overfitting.

With our features in place, we set out to build a model to predict the tackle probability of every defensive player on the field at the moment a pass is caught. Because we desired a probability, but our target variable of if the player made the tackle or not was binary in nature. Thus, we leaned into a workaround approach in which we leveraged classification models such as logistic regression, CART, random forest, XGBoost and optimal classification trees to train our model, while extracting the probabilities these classifiers also attached to each row to evaluate and present our metric. Moreover, in an example case, we fit our data to a logistic regression classifier. While the classifier predicts a 0 or 1 for each training sample, we care more about the output of the logistic function (a probability between 0 or 1) for testing purposes. This is because, due to the fact we know that exactly one person must make the tackle on each play, we take the player with the maximum tackle probability of all the defensive players on a given play and predict them to make the tackle, with every other player slated to not make the tackle. This process of selecting the maximum probability is a version of a threshold which would be enforced in typical logistic regression. At the end of this procedure, we are able to compare the predicted output on if a player makes a tackle or not to the actual outcomes in the test set.

By way of the format of our problem, our dataset was extremely unbalanced. On every given play, 10 players do not make a tackle and 1 player does. This means the true labels are skewed 91% to 0, with 9% being 1. This fact about the distribution of labels led us to believe that accuracy may not be a suitable metric for measuring the success of our model since a model which predicts all 0s would have an accuracy of 91%. For this reason, we explored other avenues of evaluating our model, finally arriving at AUC (area under the curve). The AUC metric fit our unbalanced dataset well since it gave weight to both a high true positive rate and a low false positive rate, which was not reflected in a pure accuracy calculation. Additionally, we recognize that precision and F1 score would

also have worked well in measuring success as they incorporate true positives, the smaller, yet most critical portion of the data.

We ultimately decided to train and evaluate our model using a varying set of classification models: Logistic Regression, CART, Random Forest, XGBoost, and OCT. The baseline considered and each model's best performing hyperparameters and corresponding evaluation set results are displayed in the sections below.

Baseline

Our baseline model consisted of randomly selecting one defensive player in each player to record the tackle. This led to a baseline accuracy of 84% which while seeming abnormally high, makes intuitive sense given the unbalanced nature of the data. Moreover, if the baseline predicted the wrong player to make the tackle each play, it's accuracy would be about 82% (9 out of 11 since it picked one erroneous player to make the tackle and one player who made the tackle to not make the play). Thus, the baseline benchmark of 84% is within reason. While having a high accuracy, the baseline has an abysmal AUC of 0.50 since selecting at random ensures the worst possible combination of true positive and false positive rate. In this respect, the following section will demonstrate how our various models perform significantly better than the baseline approach.

Results

After training and testing several different models by dropping insignificant, checking multicollinearity, and tuning hyperparameters, the best results for each classification model were created. Table 1 highlights the AUC, accuracy, and F1 score of each model. As mentioned earlier, F1 score was also considered as it serves as a robust indicator of overall model performance (precision and recall), especially in scenarios with imbalance class distributions.

Model	AUC	Accuracy	F1
Baseline	0.50	0.8368	0.10
Logistic Regression	0.80	0.9350	0.64
CART	0.67	0.9222	0.45
Random Forest	0.81	0.9376	0.65
XGBoost	0.82	0.9398	0.67
OCT	0.88	0.9423	0.68

Table 1: Classification Model Results

In the evaluation of the models, the optimal classification tree model emerged as the best performer, exhibiting superior performance across all metrics. Not only does it drastically improve from the baseline, but it also beats out Random Forest and XGBoost which displayed promising results. Optimal classification tree also excels in interpretability, which is always a positive when evaluating model performance. The clearer understanding of the decision-making process in combination with its superior quantitative performance positions the optimal classification tree as the most effective choice among the models considered.

To build this optimal classification, a grid search was performed to find the optimal hyperparameters. These included a minimum bucket constraint of 5, max categoric levels before warning of 15, cp of 0.00104, and entropy as the criterion. A max depth of 3 was also chosen to maintain interpretability.

Although the other models did not perform as well as the optimal classification tree, it is important to highlight the hyperparameters used. Grid search was also conducted on random forest and XGBoost across various combinations of parameters. The best hyperparameters for random forest were max depth of 20, max features of log2, minimum sample leaf of 4, and number of estimators as 150. For XGBoost, the hyperparameters used in the final model were column sample by tree ratio of 0.8, gamma of 0, learning rate of 0.1, max depth of 5, minimum child weight of 3, number of estimators as 150, and subsample of 1.

Feature selection was also performed to enhance model performance. Several features were dropped, including game information proving to be insignificant to tackle probability (quarter, down, position of offense and defense, win probability of offense and defense at moment in game, etc) as well as features with high multicollinearity (acceleration and speed of players, etc). Although there were slight deviations in features used across models, the top features for a majority of models were: rank to ball carrier, distance to ball carrier, total weight of blockers, number of blockers, difference in y coordinate, difference in x coordinate, weight of defensive player, velocity, angle to offense, and pass length. These top features make sense as the distance, speed, and obstacles towards the ball carrier are significant indicators of tackle success.

Key Impacts and Further Research

It's no secret that the adage, "defense wins championships," rings especially true in the National Football League. Of the last 13 teams with the #1 defense, every single one made the playoffs, with 5 reaching the Super Bowl. However, while the offensive side of the ball has gotten increased attention in this next generation of statistics and big data, the defense has been neglected as far as advanced analytics go. In an increasingly points-focused and pass-heavy league, "Quarterback Rating" and "Unrealized Air Yards"

are all the rage, but these metrics merely paint a picture of the offensive impact by a team. Meanwhile, rudimentary counting statistics which have existed since the dawn of the game such as “tackles” and “interceptions” are still the leading measure of impact of defensive players, creating an unbalanced representation of players abilities. In recent years, for instance, this development has given rise to the greedy cornerback, whose job it is to cover the receiver, but will prioritize getting interceptions over preventing yardage in effort to boost their numbers and earn glamor and recognition. This is the motivation which led us to create the metric of “tackle probability” to provide coaches, managers, fans and casual observers of the game with another facet to evaluate defensive players.

Moreover, we believe the possibilities “tackle probability” unlocks are boundless. Evaluators will finally have a metric which identifies intangibles such as hustle and effort as players who consistently outperform their tackle probabilities to make tackles showcase their elite determination. From coaches who can leverage this statistic to allocate playing time to general managers who need to make roster decisions during draft day or the trade deadline, tackle probability offers actionable insights for anyone involved with personnel decisions. Scouts can also leverage our model to look at college players who display an aptitude for making tackles in spite of their location on their field or obstacles in their way at the time of a catch. In a similar vein, the metric can be used by assistants and players to decide which film to prioritize by looking at areas where high tackle probabilities led to missed tackles or low tackle probabilities led to great tackles.

The tackle probability metric may also be able to be expanded to make formation decisions. Furthermore, coaches and statisticians could also apply the metric to different situations which occur pre-snap including deviations in offensive formation, the likelihood the offense will pass and the number of defenders that are lined up in the box. Looking at how these various circumstances affect tackle probabilities could offer even more insights into decision making on the football field. Down the line, the metric can even be used for prescription-based models. Further research into external factors such as a player's past season performance could augment the existing model and make an even more accurate indicator of tackle probability.

Overall, the importance of the tackle probability to football and the potential advantages it brings to stakeholders on the defensive side cannot be understated. The metric is sure to impact football in the years to come.

Partner Contributions

Both partners collaborated extensively throughout the course of this project, especially during ideation and decisions regarding scope and direction. While many major aspects of the project were completed while working together, each partner had certain individual contributions towards making the final model a success.

Krishanu dealt with large portions of data preprocessing, including cleaning all the variables, reconfiguring formatting and handling merges to bring the datasets together. He also worked on coding many of the engineered features which the team had brainstormed together.

On the other hand, Joe worked in depth on the model pipeline, ensuring the preprocessed data was free of collinear and insignificant columns before testing the data across an assortment of models. Joe spearheaded the shift of evaluation metric from accuracy to AUC as a way to deal with issues regarding the unbalanced data.

Both partners looked to one another for support and encouragement during the ups and downs of the project. The team made many breakthroughs in the project together including deciding on the process for how to evaluate the output of the various classification models. The pair also came together to discuss conclusions and takeaways from major steps along the way, including assessing tackling probability as a whole and debating the implications of applying the metric.

Appendix

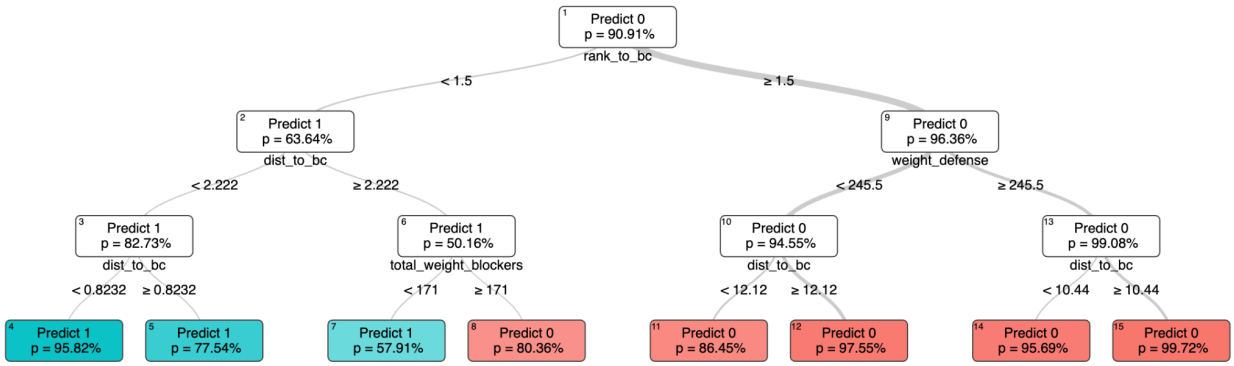

Figure 1: Optimal Classification Tree

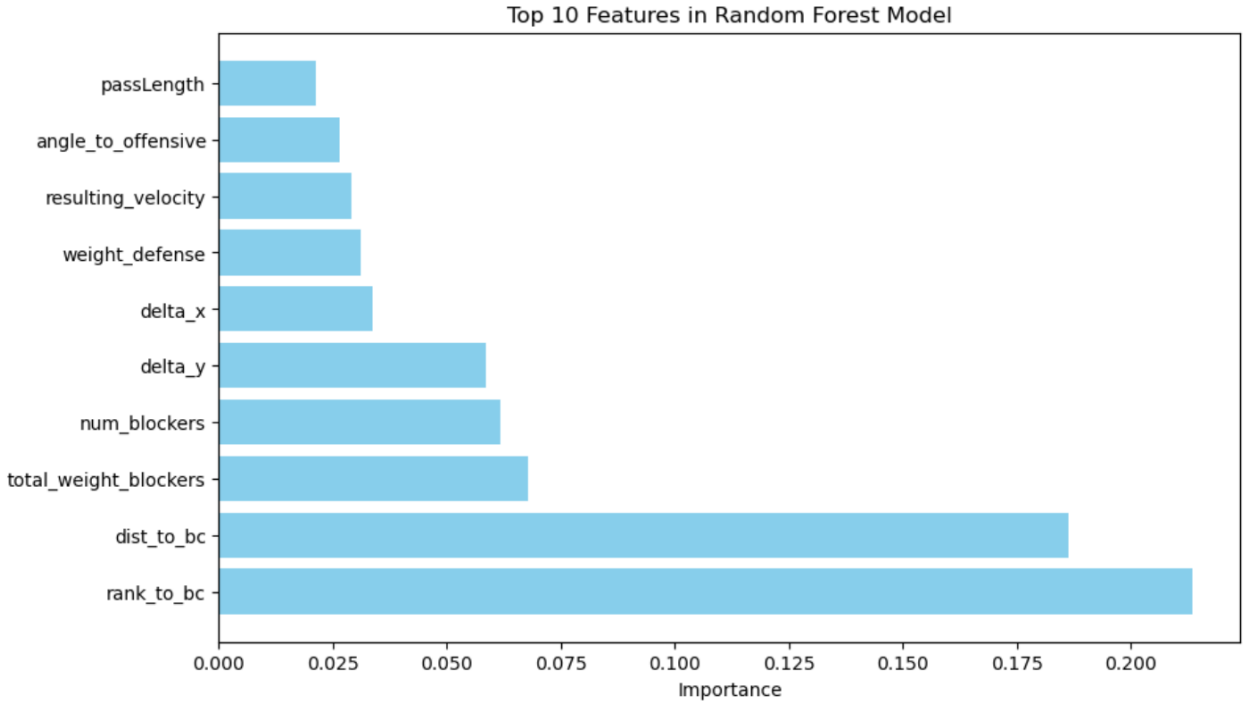

Figure 2: Top 10 Features in Random Forest Model