Predicting NFL Defensive Pass Coverages Pre-Snap with Random Forest Classification

Louis J. DiMuro
ldimuro3@gatech.edu

Abstract

This paper investigates the use of Random Forest classification to predict NFL defensive pass coverages pre-snap based on spatial/positional and game-state features. By processing 8,278 passing plays from the NFL Big Data Bowl 2025 dataset, a model was trained to predict coverages such as Cover-1, Cover-2, and Cover-3, achieving a notable improvement in baseline accuracy from 31.76% to 55.86%. Key spatial features, particularly the depth and alignment of safeties, were identified as the most influential predictors, aligning with traditional quarterback strategies for pre-snap defensive reading. However, challenges such as class imbalance affected model performance, with underrepresented coverages like Cover-6 achieving lower prediction accuracies. The study highlights the potential of spatial features and underscores the importance of class-balanced datasets for improving prediction accuracy in future models.

1 Introduction

Predicting the type of pass coverage an NFL defense will use before the snap is one of the most complex tasks in professional American football. Various factors must be considered, including the depths and alignments of defenders, their spacing, the defensive formations, the current score, and the offensive team's down and distance, among many others.

1.1 Goals

1. **Prediction of Defensive Pass Coverage**: Can an NFL defensive pass coverage (e.g., `Cover-1`, `Cover-2`, `Cover-3`, `Cover-6`, `Quarters`, or `Other`) be accurately predicted using classification modeling, based on pre-snap defender spatial/positional features and game state?

2. **Feature Importance**: What are the key features that most significantly contribute to predicting defensive pass coverage?

1.2 Hypothesis

While NFL defensive coverages are notoriously difficult to predict pre-snap, especially given the lengths defensive coordinators go to in order to disguise them, it is hypothesized that the model will be able to identify patterns in defensive positional and game-state features, leading to an improvement in baseline coverage prediction accuracy.

Specifically, it is expected that spatial and positional features such as the alignment of safeties, the depth of cornerbacks, and the number of defenders near the line-of-scrimmage, as well as game-state features such as score, down, and distance, will be the most influential. These are key factors often emphasized in quarterback training to aid in predicting defensive coverages (Nogle, 2016) [1].

2 Methodology

The goal of this project was to use 21 pre-snap features extracted from various NFL datasets, and train a Random Forest classification model to be able to classify NFL defensive pass coverage types so it may be used to make predictions in real-time.

2.1 Datasets

The data that was used for this project was taken from the NFL Big Data Bowl 2025 competition, which provided game, player, play, and tracking data from Weeks 1-9 of the 2022 NFL season (NFL Big Data Bowl 2025, n.d.) [2]. The specific datasets and data extracted from each are as follows:

2.1.1 plays.csv

This dataset contains information about every play from every game Weeks 1-9 of 2022. Since this project focused purely on passing plays, this dataset was filtered to remove all running and special teams plays. Extracted features included:

- `pff_passCoverage`: the pass coverage concept employed by the defense on the play. This value was used as the ground-truth label the model was trained to classify and predict
- `receiverAlignment`: enumerated as 0x0, 1x0, 1x1, 2x0, 2x1, 2x2, 3x0, 3x1, 3x2

- quarter
- down
- yardsToGo: distance needed for a first down
- possessionTeam: team abbreviation of team on offense with possession of the ball
- absoluteYardlineNumber: distance from end zone for possession team (values 0-120)

2.1.2 games.csv

This dataset contains basic data about every game that was played Weeks 1-9 of the 2022. The features that were extracted from each play were:

- homeTeamAbbr
- visitorTeamAbbr

2.1.3 player_play.csv

This dataset contains data about the actions that each of the 22 players performed on a given play. The features that were extracted from this dataset for each play that was processed were:

- gameId
- playId
- nflId: player identification number. Used to obtain the nflIds of all 11 defensive players and the offensive center (to obtain coordinates of the ball position)

2.1.4 tracking.csv

This dataset contains the spatial, positional, and speed data for all 22 players on the field for every play, before, at the moment of, and after the snap. Since this project focuses on pre-snap defensive coverage analysis, data was extracted for a single frame just before the snap of each play. The features that were used were:

- x: player position along the long axis of the field, 0 - 120 yards
- y: player position along the short axis of the field, 0 - 53.3 yards

2.2 Preprocessing

The goal was to obtain spatial and positional data about all 11 defenders for each passing play that was analyzed, so in terms of preprocessing once the data had been successfully extracted from each of the datasets listed above, all that was left was to combine the data into the necessary features and encode all text data to numerical values.

Since the Random Forest model used for this project can only process numerical data, it was necessary to convert all text values that were obtained to corresponding numbers. Therefore, a simple encoding of text to integers was sufficient to address this problem. For instance, the ground-truth label for this project was the pff_passCoverage of each passing play, which was sorted into the following classes: [Cover-1, Cover-2, Cover-3, Cover-6, Quarters, and Other], with Other being a consolidation of several coverages that were very rare (such as Prevent and Goal Line) and had too few samples to warrant having their own class. After encoding, these classes were converted into: [0, 1, 2, 3, 4, 5]. A similar encoding was also performed for receiverAlignment (which contains text values such as "2x2," "3x1," "2x1," etc). However some features required a bit more complex encoding.

Machine learning models like Random Forest tend to perform better when feature values are more generalized. For example, consider two plays where the offensive team is down by 12 points in one case and 13 points in another. Although both situations involve being behind by two possessions, the model might interpret these as distinct game situations due to the exact score differential. To address this, rather than using the raw score differential, the feature was transformed into a more generalized form: the possession_team_score_diff. This feature takes a positive value when the offensive team is ahead and a negative value when they are behind, capturing the broader game context of possessions rather than focusing on the exact point difference:

- score_differential $= 0$, possession_team_score_diff $= 0$
- $0 <$ score_differential ≤ 8, possession_team_score_diff $= 1$
- $8 <$ score_differential ≤ 16, possession_team_score_diff $= 2$
- $16 <$ score_differential ≤ 24, possession_team_score_diff $= 3$
- score_differential > 24, possession_team_score_diff $= 4$

A similar grouping was performed for yardsToGo:

- yardsToGo $= 1$, yards_to_go_grouped $= 0$ (Very Short/Inches)
- $1 <$ yardsToGo ≤ 5, yards_to_go_grouped $= 1$ (Short)
- $5 <$ yardsToGo ≤ 9, yards_to_go_grouped $= 2$ (Medium)
- yardsToGo > 9, yards_to_go_grouped $= 3$ (Long)

Once all the necessary data was extracted from the datasets and properly encoded, the next step was to create the spatial/positional features that were predicted to be the most useful for classifying pass defensive coverages. These features were created by leveraging the x and y-coordinates of each defender for each play. All features that were created were based on these two values for each defender:

- `defender_dist_to_los`: the distance in yards of the defender to the line-of-scrimmage, computed by:
  |`absoluteYardlineNumber - defender_x_coord`|
- `defender_y_coord`

2.3 Features

After researching what type of metrics coaches and quarterbacks primarily look for when reading a defensive pass coverage pre-snap, it appears some of the most important metrics included how deep defenders are, the spacings between them, and how many defenders are lined-up on the line-of-scrimmage (Nogle, 2016) [1].

Keeping these metrics in mind and using the two positional values described above, the following features were created:

- `avg_defender_depth`: the average `defender_dist_to_los` value across all 11 defenders
- `std_defender_depth`: the standard deviation of `defender_dist_to_los` values across all 11 defenders
- `defender_lateral_spread`: the difference between the defenders with the highest and lowest `defender_y_coord` values
- `middle_field_count`: number of defenders with `defender_y_coord` within 10 yards of the ball `y_coord`
- `outside_field_count`: number of defenders with `defender_y_coord` further than 10 yards of the ball `y_coord`
- `defenders_near_los`: number of defenders with `defender_dist_to_los` $\leq$ 1.5 yards
- `defenders_in_box`: number of defenders with `defender_dist_to_los` $\leq$ 1.5 yards and `defender_y_coord` within 10 yards of the ball `y_coord`
- `deepest_safety_depth`: defender with position listed as "`SS`" or "`FS`" with the highest `defender_dist_to_los` value
- `next_deepest_safety_depth`: defender with position listed as "`SS`" or "`FS`" with the second highest `defender_dist_to_los` value
- `safety_lateral_spread`: the difference between the highest and lowest `defender_y_coord` values among defenders with positions listed as "`SS`" or "`FS`"
- `mof_open`: "Middle of Field Open" boolean value, 1 if `deepest_safety_depth` and `next_deepest_safety_depth` are both $>$ 10 yards, 0 otherwise
- `avg_cb_depth`: the average `defender_dist_to_los` value for defenders with position listed as "`CB`"
- `min_cb_depth`: the lowest `defender_dist_to_los` value among defenders with position listed as "`CB`"
- `avg_lb_depth`: the average `defender_dist_to_los` value for defenders with position listed as "`LB`", "`MLB`", "`OLB`", or "`ILB`"
- `std_lb_depth`: the standard deviation of `defender_dist_to_los` values for defenders with position listed as "`LB`", "`MLB`", "`OLB`", or "`ILB`"
- `lb_lateral_spread`: the difference between the highest and lowest `defender_y_coord` values among defenders with positions listed as "`LB`", "`MLB`", "`OLB`", or "`ILB`"

Each play was represented by using spatial/positional and game-state features, with the `pff_passCoverage` value serving as the ground-truth label for model classification. Table 1 presents the breakdown of each ground-truth label class along with the corresponding frequency of occurrences. Overall, there were 8,278 total passing plays that were processed, with 21 input features being extracted for each play.

Class	Occurrences
Cover-3 (Class 2)	2598
Cover-1 (Class 0)	1851
Cover-2 (Class 1)	1144
Quarters (Class 4)	1079
Other (Class 5)	814
Cover-6 (Class 3)	792

Table 1: Breakdown of defensive pass coverage classes and their corresponding occurrences in the dataset.

2.4 Random Forest Model

2.4.1 Model Selection

Various machine learning models are used for classification tasks, but Random Forest was chosen for this project for several key reasons.

One significant advantage of Random Forest over other commonly used models like Naive Bayes and Logistic Regression is its ability to learn non-linear decision boundaries (AIML.com, 2024) [3]. Given the complexity of NFL defensive formations, where 11 defenders are spread across the field with distinct responsibilities, Random Forest's capability to identify patterns in seemingly noisy data was particularly valuable.

Another advantage is its ability to handle large datasets with high dimensionality (AIML.com, 2024) [3]. For this project, over 8,000 passing plays from the NFL Big Data Bowl dataset were processed to produce as many training samples as possible to get the widest range of situations and coverage looks, making Random Forest's efficiency in managing large volumes of data essential for achieving high performance.

Lastly, Random Forest provides insights into feature importance (AIML.com, 2024) [3], which is crucial for understanding which features contribute most to predicting defensive coverage pre-snap. This ability to assess and quantify feature significance helps prioritize the most relevant aspects of the data when training the model.

2.4.2 Hyperparameters

There were three hyperparameters that were tuned to achieve the highest possible performance:

- `n_estimators`: Random Forest is an ensemble model that combines multiple decision trees (Random Trees) to make predictions. Each tree casts a prediction "vote," and the class with the most votes is selected for prediction. Higher values of `n_estimators` generally improve model stability and reduce susceptibility to data variations, but at the cost of increased computational time (Saxena, 2024) [4]
- `max_depth`: deeper trees in Random Forest can model more complex relationships between features, but excessively deep trees may lead to overfitting, where the model memorizes the training data and fails to generalize well to new data (Saxena, 2024) [4]
- `class_weight`: as shown in Table 1, coverages such as `Cover-1` and `Cover-3` have more instances than `Quarters` and `Cover-6`, leading to class imbalance. This can cause the underrepresented classes to perform poorly. To address this, the `class_weight` was set to "`balanced`," which prioritizes underrepresented classes during training

To determine the optimal `n_estimators` and `max_depth` values, four Random Forest models were trained with `max_depth` values of 10, 20, 30, and 40. The accuracy of each model was evaluated across a range of `n_estimators` from 10 to 500 (in increments of 10). Figure 1 presents the accuracy curves for these models. The model with `max_depth`=30 achieved the highest overall accuracy, peaking at `n_estimators`=420 before gradually declining. Therefore, the final hyperparameter values selected were `max_depth`=30 and `n_estimators`=420.

2.4.3 Training and Validation

After processing all 8,278 plays with the appropriate features and labeling them with the corresponding `pff_passCoverage`, the data was ready to be input into the Random Forest model for training. The data was split into a training set (80%, or 6,622 plays) and a validation set (20%, or 1,655 plays). The training set was used to train the Random Forest model, while the validation set served to evaluate the model's performance on unseen data.

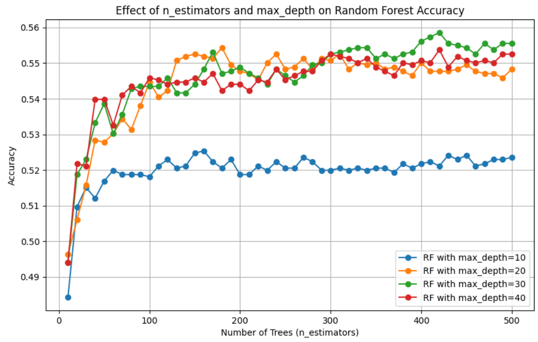

Figure 1: Effect of `n_estimators` and `max_depth` on Random Forest model accuracy

2.4.4 Evaluation Metrics

Several metrics are used to evaluate the performance of the Random Forest model:

- **Accuracy**: the ratio of correctly predicted defensive coverages to the total number of predictions. This metric provides a general measure of model performance
- **Precision**: the number of correctly identified instances of a specific defensive coverage divided by the total number of times the model predicted that coverage. This indicates how often the model's predictions for a given coverage were correct (Kreiger, 2021) [5]
- **Recall**: the number of correctly identified instances of a defensive coverage divided by the total actual instances of that coverage in the dataset. This measures the model's ability to capture all relevant instances of each coverage type (Kreiger, 2021) [5]
- **F1-Score**: The combination of precision and recall, offering a balanced measure of model performance by considering both the ability to correctly predict coverages (precision) and the ability to identify all relevant instances (recall) (Kreiger, 2021) [5]

In this project, recall is the most important metric, as it evaluates how well the model identifies all instances of each coverage type, particularly for minority classes like `Cover-6`, which may be missed by the model. The F1-score is also critical because it balances both precision and recall, providing a more comprehensive assessment of model performance across all coverage types. While accuracy can be informative, it may be misleading in imbalanced datasets, as a model could achieve high accuracy by simply predicting the majority class. Therefore, a combination of Recall and F1-score offers a more reliable evaluation, ensuring the model's ability to predict coverages accurately and completely, despite the class imbalances in the dataset.

3 Key Findings & Insights

After utilizing all 8,278 processed passing play instances, each with 21 input features, the Random Forest model was able to create a substantial improvement on the baseline accuracy of 31.76% (the accuracy of the model if it were to randomly guess the coverage of every play in the validation set), improving it to 55.86% (a 75.9% increase). The model's performance was evaluated using several metrics, including precision, recall, and F1-score. The weighted average recall and weighted average F1-score of all coverages were 56% and 55%, respectively, indicating that the model was able to identify underlying patterns within features of the data. The most important features that contributed most to the model performance were all depth-related, with the top most important features being `deepest_safety_depth` and `next_deepest_safety_depth`, with game-state features like `yards_to_go` and `receiver_alignment` all contributing very little. Despite the increase in performance from the baseline, it is apparent that predicting a defensive coverage pre-snap using only static spatial/position and game-state features is a tough task.

3.1 Model Performance Analysis

Table 2 presents the performance results of the model, with precision, recall, and F1-score for each of the 6 coverage classes, as well as the weighted averages of each category across all coverage classes.

Class	Precision	Recall	F1-Score
Cover-1 (Class 0)	0.65	0.64	0.65
Cover-2 (Class 1)	0.42	0.33	0.37
Cover-3 (Class 2)	0.57	0.72	0.63
Cover-6 (Class 3)	0.42	0.33	0.37
Quarters (Class 4)	0.49	0.41	0.45
Other (Class 5)	0.64	0.56	0.60
Accuracy		0.56	
Weighted Avg	0.55	0.56	0.55

Table 2: Precision, recall, and F1-score for each coverage class along with Accuracy and Weighted Average

The highest precision and recall values were achieved by `Cover-1` (Class 0) and `Cover-3` (Class 2), likely due to the large number of instances available for these coverages, giving the model more examples to learn from and improve its performance. In contrast, coverages like `Cover-2` (Class 1), `Cover-6` (Class 3), and `Quarters` (Class 4) had significantly fewer instances, which likely contributed to their lower precision, recall, and F1-scores.

Interestingly, `Other` (Class 5), despite having the second-lowest number of instances, performed better than expected, producing the third-highest results in terms of F1-Score. This is likely because the `Other` category includes rare and distinct coverages like `Prevent` and `Goal Line`, which have unique spatial and positional characteristics. For example, `Prevent` coverage involves defenders positioned very far from the line-of-scrimmage, while `Goal Line` coverage has defenders positioned very close to it. These distinctive features most likely made these coverages easier to identify, even with fewer occurrences.

The weighted average was preferred over the macro average because the weighted average accounts for the distribution of class sizes, making it a more accurate reflection of model performance in real-world scenarios (scikit-learn, n.d.) [6]. Since `Cover-6` (Class 3) is a less common coverage and `Cover-1` (Class 0) is more frequent, treating them equally would not reflect the true impact of class imbalance in real-world situations. The weighted average values for precision, recall, and F1-score (55-56%) indicate moderate overall performance, with a slight emphasis on correctly identifying the more frequent coverages. The close alignment between precision and recall suggests that the model is not overemphasizing either completeness or accuracy, although there is room for improvement, particularly for underrepresented classes like `Cover-6` and `Quarters`. The low scores for these coverages underscore the challenges posed by class imbalance in machine learning models, where rare classes are harder to predict accurately.

3.2 Confusion Matrix

A confusion matrix is a table that evaluates a classification model's performance by comparing its predictions to the actual results (GeeksforGeeks, 2025) [8]. In this case, it shows how the Random Forest model's predicted coverages align with the true coverages assigned to each play. Figure 2 presents the confusion matrix for the coverages `Cover-1`, `Cover-2`, `Cover-3`, `Cover-6`, `Quarters`, and `Other`.

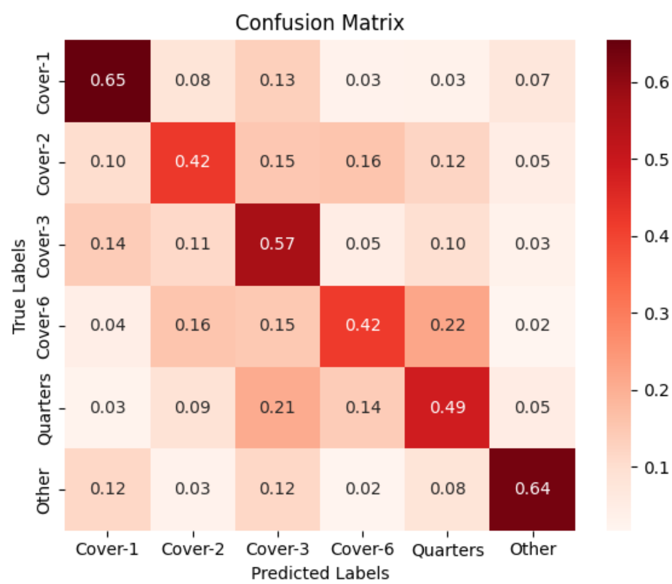

Figure 2: Confusion matrix comparing predicted and actual pass coverages

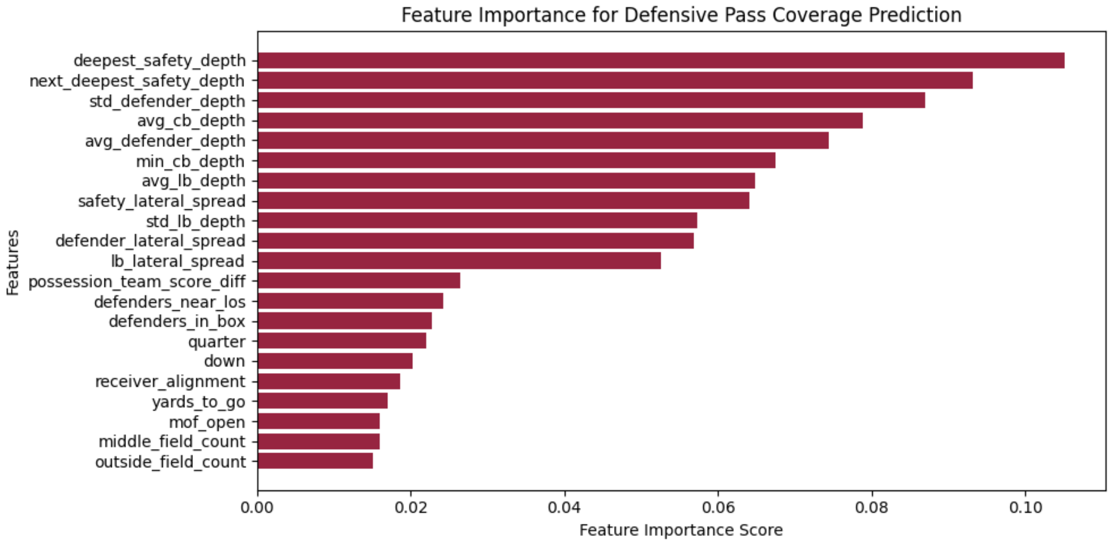

Figure 3: Feature Importance plot highlighting the most influential features in predicting defensive pass coverage

A well-performing model should produce a confusion matrix where the diagonal values contain the highest precision scores, indicating correct predictions of the actual class label. As shown in Figure 2, while not perfect, the diagonal values are the darkest and show the highest precision values, with the model correctly predicting the true label for each coverage most of the time. The model performed moderately well for `Cover-1`, `Cover-3`, and `Other`, correctly predicting 57%-65% of instances. However, it struggled with `Cover-2`, `Quarters`, and especially `Cover-6`, where it correctly predicted only 42%-49% of instances. This was expected, as these coverages were less represented in the dataset, with `Cover-6` having the smallest class size and the lowest Precision at 42%. This highlights the impact of class imbalance on model performance, and these low precision values could potentially improve with more instances of underrepresented coverages like `Cover-6` and `Quarters`.

3.3 Feature Importance

One of the key reasons for using the Random Forest model in this project is its ability to quantify the contribution of each input feature to model performance. Figure 3 displays the importance of each feature for predicting defensive pass coverage pre-snap from most important to least important.

These results clearly indicate that depth, particularly the alignment of safeties, is the most important factor for predicting defensive coverage. The features `deepest_safety_depth` and `next_deepest_safety_depth` ranked as the top two, which aligns with common pre-snap reading techniques used by quarterbacks (Nogle, 2016) [1]. One of the first ways quarterbacks are taught to diagnose a defensive pass coverage pre-snap is identifying how many safeties are dropped back deep from the line-of-scrimmage (Jenkins, 2019) [7].

The importance of safety depth outweighs that of cornerback (CB) and linebacker (LB) depth, with safeties being the most influential, followed by cornerbacks and then linebackers, as shown in Figure 3. The lateral spread of defenders appears less important than their depth, as these features rank lower in importance. However, position-based spatial features, such as depth and spread, are significantly more important than game-state features (e.g., `down` and `yards_to_go`) or other spatial features (e.g., `defenders_in_box` and `defenders_near_los`). This is evident from the large drop-off in Feature Importance Score between the lowest position-based spatial feature (`lb_lateral_spread` with a score of 0.0526) and the next highest feature (`possession_team_score_diff` with a score of 0.0264). Interestingly, `defenders_in_box` and `defenders_near_los`, despite being key metrics in identifying pass coverages for quarterbacks (Nogle, 2016) [1], had relatively low scores.

4 Further Improvements

While it's challenging to predict pass coverage with certainty based solely on defenders' static positions pre-snap, this model has demonstrated that underlying patterns can be leveraged for prediction. However, there are opportunities to enhance its performance.

Two key methods often used by quarterbacks to identify coverages involve motion. First, a quarterback can send a receiver "in motion" by having them move along the offensive formation before starting the play. Observing the defense's reaction to the receiver's movement can offer insights into the coverage they are about to play (Nogle, 2016) [1]. Second, defenders often begin moving in relation to their coverage assignment before the snap. For instance, a defender who is about to blitz the quarterback might begin moving toward the line-of-scrimmage moments before the ball is snapped. Incorporating dynamic features, like defender movement in response to receiver motion and defender acceleration just before the snap, could significantly improve the model's accuracy by adding more context to the defenders' pre-snap positioning.

5 Conclusion

Overall, there were several key findings that were uncovered through the process of this project.

Firstly, the model's ability to increase the baseline accuracy from 31.76% to 55.86% (a 75.9% increase) supports the first part of the hypothesis that underlying patterns in pre-snap defender spatial features and game-state features can be classified by a machine learning model and leveraged to predict defensive pass coverage.

The second part of the hypothesis was partially validated. It was predicted that commonly taught techniques, such as observing the alignment of safeties, cornerback depth, number of defenders in the box, as well as game-state features like score, down, and distance, would be the most influential. The model confirmed that safety positioning (with the depth of safeties being the top two most important features) is critical. However, contrary to expectations, the depth of defenders proved to be the most significant factor, while features like the number of defenders in the box and game-state had much less impact on predictions, challenging conventional methods.

The final key insight was the impact of class imbalance on model performance. Coverages such as `Cover-1` and `Cover-3`, which had a higher frequency in the dataset, achieved the highest prediction accuracies. In contrast, `Cover-6` and `Quarters`, which were less frequent, yielded the lowest accuracies due to insufficient training data. This underscores the importance of a well-balanced dataset for training machine learning models and suggests that more data from underrepresented coverages could improve accuracy and overall model performance.

References

[1] Nogle, K. (2016, June 27). Football 101: How a quarterback "reads" a defense (aka defensive coverage schemes). *The Phinsider*. https://www.thephinsider.com/2016/6/27/12039106/football-101-defensive-cover-schemes-aka-how-a-quarterback-reads-a-defense

[2] NFL Big Data Bowl 2025. *Kaggle*. (n.d.). https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/data

[3] AIML. (2024, May 22). What are the advantages and disadvantages of random forest? https://aiml.com/what-are-the-advantages-and-disadvantages-of-random-forest/

[4] Saxena, S. (2024, December 17). A beginner's guide to random forest hyperparameter tuning. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/#h-random-forest-hyperparameter-1-nbsp-max-depth

[5] Kreiger, J. (2021, September 10). Evaluating a random forest model. *Medium*. https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56

[6] F1_score. *scikit*. (n.d.-a). https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[7] Jenkins, T. (2019, February 10). How to teach quarterbacks to read defenses. *Jenkins Elite*. https://jenkinselite.com/how-to-teach-quarterbacks-to-read-defenses/

[8] GeeksforGeeks. (2025, January 15). Understanding the confusion matrix in machine learning. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/