Predicting Ideal Play Calling and Momentum Metrics Based on Real-Game NFL Play-By-Play Data

Team 1

Members: Alexander Kosecki, Niko Montez

Sept 8, 2024

Executive Summary:

This project explores the use of predictive analytics in football to enhance play-calling strategies and understand game dynamics. Leveraging the comprehensive NFLverse dataset, which includes play-by-play information and advanced metrics, we developed machine learning models to analyze three key areas: the effectiveness of routes against specific defensive coverages, optimal defensive strategies against offensive formations, and a quantifiable metric to track game momentum. These analyses aim to provide NFL teams with actionable insights for data-driven decision-making and improved on-field performance.

For our first research question, we identified optimal offensive routes against defensive coverages by using a K-Nearest Neighbors (KNN) model. The findings revealed that routes such as slants and screens are highly effective against man-to-man and prevent defenses, respectively. The second research question focused on determining which defensive coverages work best against offensive formations. Using Random Forest and Gradient Boosting models, we found that strategies like Cover 3 performed well against formations such as trips. The final research question addressed game momentum, where we developed a metric based on Expected Points Added (EPA), Win Probability Added (WPA), and other factors. Random Forest models provided the most accurate momentum predictions, highlighting significant game events like turnovers and scoring plays as key momentum indicators.

While the results showed moderate model accuracy (e.g., 50% for RQ1 and 40% for RQ2), they reflect the inherent variability and complexity of football. Nevertheless, the models identified critical trends and patterns that coaches and analysts can incorporate into their strategies. Ethical considerations include addressing potential biases from historical data,

ensuring equitable access to analytics tools, and preventing over-reliance on models that may oversimplify the game's complexity. These insights should enhance decision-making without replacing human expertise.

Challenges included handling class imbalances, cleaning large datasets, and creating a momentum metric from intangible factors. Despite these hurdles, the results demonstrate the potential of predictive analytics to improve football strategies. We recommend that NFL teams integrate these insights into play-calling and defensive preparation while continuing to refine models by incorporating additional variables, such as environmental factors. Our work underscores the value of combining advanced analytics with expert judgment to maximize team performance.

Table of Contents

Project Plan

Organization Description:

The National Football League is a professional American football league comprising 32 teams divided between the National Football Conference and the American Football Conference. Founded in 1920, the NFL headquarters are located in New York City and is widely regarded as the premier American football league in the world.

The league operates with a 17 game regular season that typically runs from September to January, culminating in the playoffs and the Super Bowl, which determines the league champion. The NFL is known for its intense level of competition, extensive media coverage, and significant cultural impact in the United States. It also engages in various community initiatives and has a global presence through international games and partnerships.

Research Questions:

RQ1: What routes are ideal against certain defensive coverages?

As teams seek to improve their offensive strategies, it is crucial to understand which routes are most effective against specific defensive coverages. The ability to identify and exploit defensive schemes can significantly enhance a team's passing game, making the offense more unpredictable and difficult to defend. For example, recognizing when a defense is in Cover 2, Cover 3, or man-to-man can allow quarterbacks and receivers to adjust their routes, such as running quick slants against man coverage or deep posts against Cover 2, to maximize yardage and create scoring opportunities. Ideal routes are those that capitalize on the weaknesses inherent

in a particular coverage, such as attacking the deep middle in Cover 3 or exploiting the flats in Cover 4.

The aim of this research question is to identify the attributes of specific routes that closely match predictable patterns in defensive schemes, providing a tactical advantage to the offense. By understanding these matchups, offensive coordinators can design game plans that target vulnerabilities in the defense, ultimately improving the team's overall performance on the field.

RQ2: What defensive coverages are ideal against certain offensive formations?

Similarly to the offensive side, a key component of defensive success is understanding which defensive coverages are the best based on the information that the offense is giving in the form of formation. For example, if the defense can recognize with a high likelihood that the offense will be attempting a run based on the offensive formation and previous history in those formations, they would be given a great resource to at least know to look out for the predicted play or even further, use the predicted coverage call. This would allow the defense to exploit the information given by the offensive formation and be given the best predicted play call to make in real time.

The goal for this research question is to identify the indicators in formation that most often result in certain offensive plays, giving an advantage to the defense by gaining this knowledge. As mentioned, with this knowledge the defensive coordinators can call defensive plays that are made to stop the given play, resulting in a higher percentage of defensive stops.

RQ3: How can we create a metric to track the momentum of a given game?

As sports analytics continue to evolve, developing a reliable metric to track the momentum of a game is increasingly important for teams, analysts, and fans. Momentum, though often

considered intangible, can significantly influence the outcome of a game, affecting player confidence, strategic decisions, and overall team performance. For example, a sudden shift in momentum after a key turnover or a scoring run can dramatically alter the dynamics of a game, making it crucial to quantify these shifts in real time.

The objective of this research question is to establish a metric that captures the ebb and flow of momentum by analyzing factors such as scoring runs, turnovers, and time of possession. By identifying the attributes that align with changes in momentum, this metric could provide valuable insights into game dynamics, enabling teams to make data-driven decisions that capitalize on or counteract these critical shifts.

Hypotheses

H1: Specific routes and combinations of routes will be optimal against certain defensive coverages.

By analyzing NFL play-by-play data, we hypothesize that specific routes, such as slants, posts, and outs, will be most effective against particular defensive coverages like Cover 1, Cover 2, and other defensive formations. By identifying these optimal route-coverage matchups and examining their success rates in terms of yards gained, completion percentages, and touchdowns, we will be able to test this hypothesis.

H2: Certain defensive coverages will be optimal against given offensive formations.

Using the play-by-play data along with a dataset of offensive and defensive formations, we hypothesize that certain defensive coverages will prove to be the most effective against the formation that the offense has come out in such as Bunch or Twins. We will test this hypothesis

by identifying the optimal coverage based on the lowest stats in terms of resulting yards gained, completion percentages and touchdowns.

H3: Changes in advanced metrics can track momentum shifts during games.

We propose that shifts in advanced metrics, such as expected points added (EPA) and win probability, can indicate momentum changes during games. By identifying key plays and their impact on these metrics, such as turnovers, scoring drives, or defensive stops, we can create a reliable metric to track momentum. This hypothesis will be tested by analyzing the correlation between game events and changes in momentum, validating the approach against game outcomes and expert observations.

Data

The data comes from the NFLverse dataset, which contains detailed play-by-play information from NFL games. This dataset includes both structured and unstructured variables that describe each play, allowing for in-depth analysis of football strategies. The dataset captures data from regular season and postseason games, offering insights into various aspects of team performance, player actions, and game dynamics. The data is divided into several key modules, including play descriptions, game context, player statistics, and advanced metrics.

Play Attributes

Each play is described with variables such a play type (pass, run, etc.), coverage type, offensive formation, route combinations, down, distance, yards gained, and game situation. Additional attributes include offensive and defensive team identifiers, player names, play results (e.g., completions, turnovers), and penalties. This information allows analysts to explore which routes

work best against specific coverages and which defensive setups counter certain offensive packages effectively.

Game Context and Advanced Metrics

The dataset provides detailed game context, including quarter, time remaining, score differential, and field position. Advanced metrics include expected points added (EPA), win probability, and success rates, which help assess the impact of each play on the game's outcome. These metrics can be instrumental in developing a momentum tracking metric by analyzing shifts in EPA, sudden changes in win probability, and scoring sequences.

Coverage and Formation Data

Key variables for studying defensive and offensive strategies include the type of defensive coverage (Cover 1, Cover 2, man-to-man, zone) and offensive formation data (trips, I-formation, shotgun). By analyzing how these factors interact, researchers can identify optimal defensive strategies against specific offensive packages and vice versa.

Measurements

Critical measurements in this analysis include the effectiveness of route combinations against particular defensive coverages, the success rate of different defensive schemes against offensive formations, and the development of a momentum metric based on game flow data. Effectiveness can be measured by yards gained, first down conversion rates, and EPA per play, while defensive success can be quantified by metrics like forced turnovers and stops.

For momentum tracking, the analysis focuses on capturing shifts in game dynamics, such as changes in win probability, scoring runs, and significant plays that impact the game's flow. This metric aims to quantify the often-intangible feeling of momentum, providing coaches and analysts with actionable insights during games. The NFLverse dataset, with its detailed and expansive data, serves as the foundation for these analyses, enabling the exploration of the relationships between plays, strategies, and game outcomes.

Methodology

For research question 1, we are trying to determine which routes are optimal against specific defensive coverages. To achieve this, we will use structured data from the NFLverse dataset such as play type, route run, and defensive coverage. A series of classification models, such as Decision Trees and Random Forests, will be used to predict the success of each route against various coverages. We will begin by performing exploratory data analysis (EDA) to identify key variables that contribute to the success of a play, such as yards gained and completion percentage. Visualization of route effectiveness against specific coverages will also be employed to validate our findings.

For research question 2, which seeks to identify the optimal defensive coverages against certain offensive packages, we will utilize structured data focusing on offensive formations and defensive schemes. Using logistic regression and Support Vector Machines (SVM), we will analyze the effectiveness of each defensive coverage in stopping specific offensive packages. The analysis will leverage structured data on play outcomes, such as turnovers and stops, to predict which defensive strategies are most successful against various offensive setups. Social

Network Analysis of team performance patterns may also provide insights into how defenses adapt to offensive schemes.

For research question 3, which aims to create a metric to track momentum during games, we will focus on advanced structured data that captures game events, such as scoring plays, turnovers, and shifts in win probability. For this problem, time series analysis will be critical, but it will also be important to analyze sequential play data to capture momentum swings. Key plays will be evaluated using Hidden Markov Models or neural networks to detect patterns indicative of momentum changes. By identifying these shifts and their impact on the game's flow, we can develop a composite momentum score that accurately reflects in-game dynamics.

For structured data, we will perform EDA to identify the most relevant variables, such as success rates of routes, defensive stops, and changes in win probability. Plotting these variables will help us understand their influence on game outcomes. In some cases, common football knowledge will be applied; for instance, understanding that aggressive defensive plays might shift momentum more significantly than standard stops.

Overall, these approaches will combine statistical modeling, machine learning techniques, and domain knowledge to address each research question effectively, leveraging the data within the NFLverse dataset to optimize football strategies and track game momentum.

Computational Methods and Outputs:

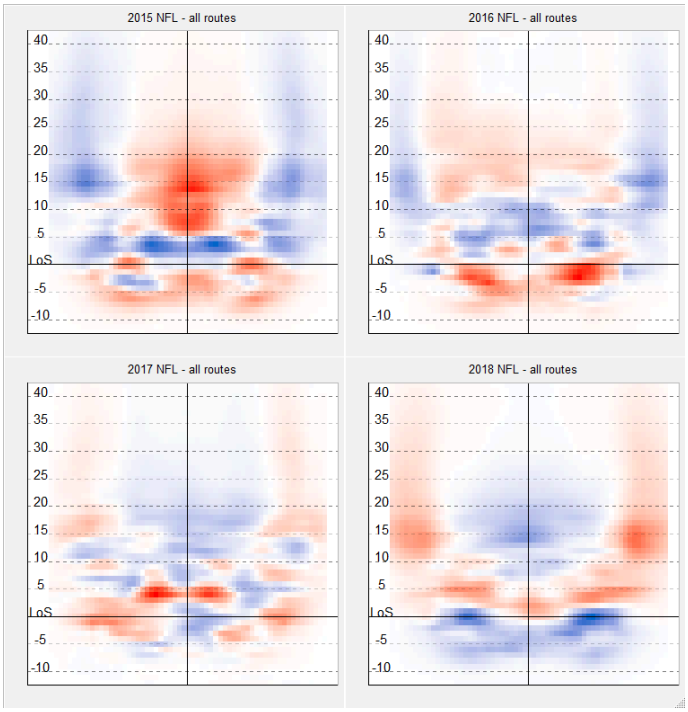
We believe that using AUC/ROC to evaluate model performance will yield the most accurate models for RQ1 as well as RQ2 and RQ3. In tandem with this, we will also be evaluating using MSE and RMSE as a way to diversify model performance evaluation. These metrics will help

assess how well the model performs in practical scenarios where certain types of predictions are more important than others. In order to tune the models, RQ1 and RQ2 will be tuned using feature selection as well as multi-fold cross validation. As for RQ3, we will be using a time-based cross-validation, as the momentum of a game is critically based on the timing of events. This will allow us to maintain the temporal order and remove the risk of future data leakage.

Output Summaries:

RQ1: What routes are optimal against certain defensive coverages?

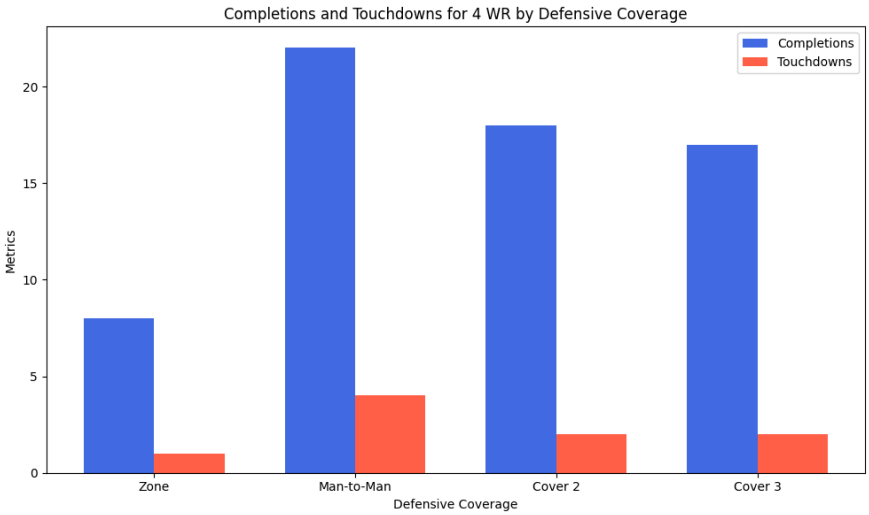
The analysis will identify the routes that have the highest success rates against specific defensive coverages. A table will be generated displaying the top 25 route-coverage combinations sorted by highest yards gained and completion percentage. Additional tables will break down the top 10 optimal routes for each common coverage type (e.g. Cover 1, Cover 2, man-to-man). A heat map will visualize these optimal routes on the field, highlighting areas where routes are most successful against various coverages. This will help illustrate the spatial effectiveness of different routes under specific defensive schemes.



https://www.pff.com/news/pro-pff-data-study-examining-the-passing-game-with-route-heat-maps

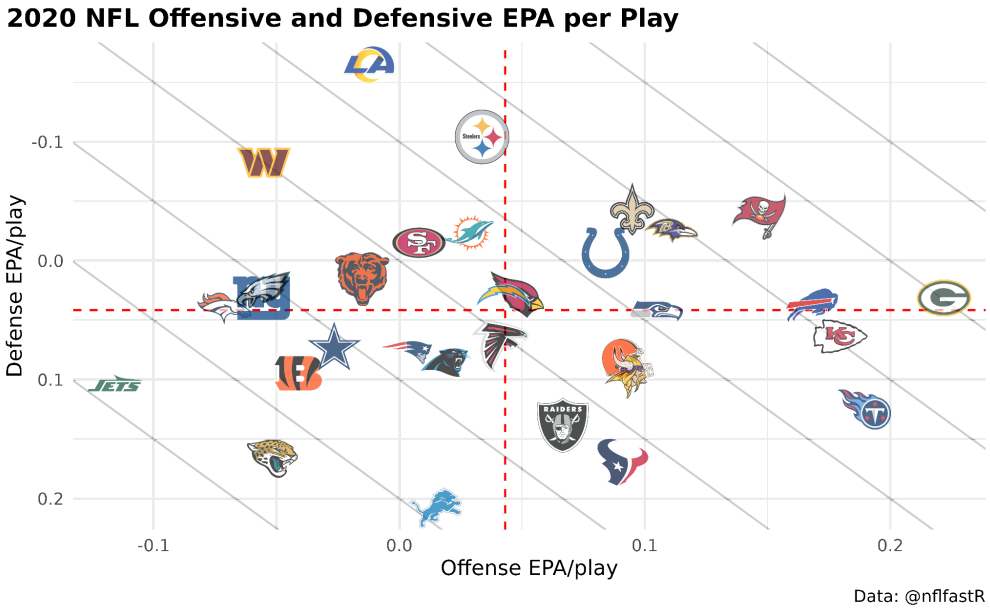
RQ2: What coverages are optimal against certain offensive packages?

The analysis will determine the most effective defensive coverages against specific offensive packages. A summary table will list the top 10 defensive coverages sorted by their success rates against offensive formations such as shotgun, trips, and I-formation. Additional output will include a matrix that pairs each offensive package with its most effective countering coverage. A pie chart will illustrate the distribution of defensive success by formation type, and a bar chart will compare the frequency of defensive stops, turnovers, and other key outcomes across different coverages.



RQ3: How can we create a metric to track momentum during games?

The analysis will develop a momentum metric that quantifies game flow and momentum shifts based on key play data. The output will include a time series graph that tracks the momentum score throughout a game, highlighting key plays that trigger significant shifts. A table will summarize the momentum change for critical moments in various games, such as touchdowns, turnovers, and defensive stops. A scatterplot will be used to compare changes in win probability against the new momentum metric to validate its effectiveness. The momentum metric will also be visualized in a line graph, showing how it fluctuates in correlation with game events, providing a clear visualization of momentum trends within a game. Additionally, we can utilize the graphing tools within NFLverse to plot different teams momentum value during games compared to winning probabilities.



https://nflplotr.nflverse.com/articles/nflplotR.html

Campaign Implementation:

The National Football League brings in 18.6 billion U.S dollars annually, and continues to reach new viewership highs every year. The 2024 Super Bowl averaged 120.3 million viewers on CBS alone, making it the largest audience for a single-network telecast to date. The Next Gen Stats tracking system records player data—including location, speed, distance traveled, and acceleration—at a rate of 10 times per second, tracking movements with precision down to inches. This raw data is utilized to automate player participation reports, compute performance metrics, and generate advanced statistics through machine learning on AWS. Each play in every game generates over 200 new data points. As the use of Machine Learning continues to grow, other organizations such as NCAA Football could begin to use metrics such as the ones used in our research questions to create real time suggestions for coaches and players. For example,

developing a model to answer the question, “How can we create a metric to track the momentum of a given game?” would allow a team to quantify momentum shifts in real time and change the game plan accordingly.

Literature Review

As predictive analytics become increasingly embedded in sports, the NFL has seen a surge in the application of machine learning and statistical modeling to improve decision-making on the field. This literature review explores studies that focus on predicting ideal play calling and tracking momentum using real-game NFL play-by-play data, highlighting the methodologies and findings that contribute to a deeper understanding of football strategy. The effectiveness of specific offensive routes against defensive coverages has been a critical area of research. In the Stanford University study on predictive modeling in NFL play outcomes, researchers utilized deep learning techniques to analyze player tracking data and evaluate the success of various routes. They concluded that “predictive models can provide strategic insights into which plays are likely to succeed against different defensive setups, allowing teams to tailor their approach dynamically” (Stanford, 2020). This study underscores the potential of using data to match offensive routes, like quick slants or deep posts, to specific defensive coverages such as man-to-man or Cover 2.

Further supporting this, the Oxford Academic study employed hidden Markov models to predict play calls based on game situational data, demonstrating that routes and formations significantly impact the success of offensive plays. The research highlighted that “routes like slants or posts are particularly effective against man-to-man coverage due to their ability to create quick separation” (Oxford, 2021), thus providing empirical backing for route-specific play

calling strategies. These findings align with broader trends in sports analytics, where understanding the matchup between offensive routes and defensive alignments is crucial for maximizing play success. The use of detailed data allows teams to not only predict but also exploit these matchups, creating a significant tactical advantage.

Understanding which defensive coverages best counter specific offensive formations has also been extensively studied. A paper published by IOS Press examined various machine learning models, including neural networks and random forests, to predict play types based on situational data. The researchers found that “defensive schemes like Cover 3 showed higher success rates against formations like trips, due to their balanced approach in covering deep threats and underneath routes” (IOS Press, 2024). This finding demonstrates the value of adapting defensive coverages to the offensive setup, enhancing a defense’s ability to counteract strategic offensive plays. Additionally, the study emphasized the importance of situational awareness and feature selection, stating that “incorporating variables such as down, yards to go, and score differential significantly improved model accuracy, providing a nuanced approach to play prediction that better informs defensive play-calling” (IOS Press, 2024). This aligns with the broader theme of integrating contextual game factors to refine defensive strategies and emphasizes the dynamic nature of football, where both offensive and defensive coordinators must constantly adjust based on in-game conditions.

Momentum in sports has very recently begun to be explored as it is complex to quantify an intangible feeling such as a momentum swing in a sporting event. For the purpose of this review, momentum is defined as a team's time-based probability of winning the match. An article from the *2nd International Conference on Artificial Intelligence, Database and Machine*

Learning (AIDML 2024) established a time predictive model based on the scoring timeline by using Hidden Markov models. They concluded that “momentum impact factors and player performance, … variables such as serve_no, point_victor, p1_winner, winner_shot_type, p1_net_pt_won, p1_distance_run, rally_count, and speed_mph exhibits strong correlations with momentum”(Jia, Z. and Li , Z. 2024), opening the door to the idea that the exact same type of momentum metric can be created and studied for the National Football League. If certain metrics that impact a teams performance can be used in a time sensitive model that takes into account the times of each action in relation to the team's ability to score points, a so-called “momentum metric” could be created. The article goes on to explain how “coaches and players can formulate strategies for serving selection and tactical arrangements based on the information provided by the model to maximize the utilization of momentum changes and develop targeted strategies against opponents' weaknesses and habits”(Jia, Z. and Li , Z. 2024). This concept as well lends itself perfectly to the idea that the same type of actions could be an option for NFL coaches and players. By tailoring their gameplay to attack harder at a positive momentum swing, or become far more conservative for a negative one, a team could potentially be at a great advantage by using a system such as this.

Exploratory Data Analysis

For our project we decided to merge two different datasets that we had found because each dataset has its own set of metrics that will prove to be useful in our data analysis and model creation. Both of these datasets come from the NFLVerse ecosystem in R. This ecosystem is a comprehensive combination of packages and repositories centered around real game NFL data with records going back decades. Each dataset had about 50,000 observations and 390 different

variables. Due to the extensive amount of variables in the extremely detailed data, we decided to clean the merged dataset by only using the following variables for the remainder of the project:

RQ1: What routes are ideal against certain defensive coverages?

Play_type - Filters out if the play is a pass or a run

Defenders_in_box - Displays the defensive front

defense_coverage_type - Core to identify how defenses respond to offensive formations.

Pass_length - Measures depth of passes

Pass_location - Where on the field the pass was attempted (left, right, middle)

Air_yards - Distance ball traveled in the air

Route - Possible alternative to pass_route

Epa - Expected points added estimates the success of each play

RQ2: What defensive coverages are ideal against certain offensive formations?

defense_coverage_type - Core to identifying how defenses respond to offensive formations.

defense_personnel - Describes defensive personnel on the field.

offense_formation - Describes the offensive setup, which the defense is reacting to.

Offense_personnel - Important to see which players are on the field and how defenses respond.

Defenders_in_box - Helps analyze how many defenders are committed to stopping the run.

number_of_pass_rushers - Indicates defensive pressure strategy.

Players_on_play - Total number of players involved, useful for defensive and offensive alignment analysis.

run_location - Helps understand how the defense reacted to run plays.

run_gap - Important for evaluating whether defensive alignment covers the run effectively.

sack - Key to knowing whether defensive coverage succeeded in pressuring the quarterback.

interception - Represents defensive success in coverage.

penalty_type - Helps analyze whether defensive coverages are prone to penalties.

RQ3: How can we create a metric to track the momentum of a given game?

score_differential - Reflects the current score margin, which is crucial for momentum shifts.

td_prob - Probability of scoring a touchdown, can be an indicator of momentum in a drive.

fg_prob - Field goal probability, indicating potential scoring opportunities.

Wpa - Win probability added, a metric that reflects shifts in game momentum.

Epa- Expected points added, which can indicate momentum in terms of offensive production.

yards_gained - Measures the success of plays and indicates shifts in momentum.

drive_ended_with_score - Whether a drive resulted in a score, affecting momentum.

turnover - Any turnover can represent a significant momentum shift.

total_home_score - The current score of the home team, essential for momentum tracking.

total_away_score - The current score of the away team, also critical for momentum tracking.

Along with these variables, the objective for answering our third research question is to create a final variable;

Momentum - Identifies the momentum of the given team based on a combination of scoring probabilities, game states, and dramatic shifts such as turnovers or big plays.

Exploratory Plotting for RQ1

To begin with our first research question we first wanted to start by plotting the yards gained grouped by the defensive coverage type for each route in our dataset. This gives us an initial look into the relationship between variables that will be very important when creating our model.

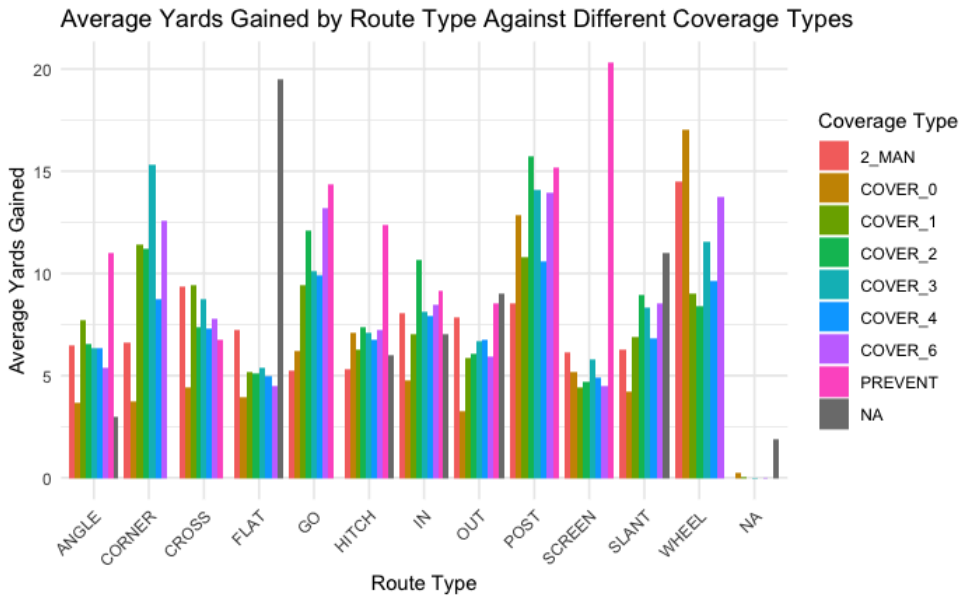


Figure 1

We can already begin to see some of the best routes against certain coverages based on the averages yards gained by those routes. The most visible of these observations can easily be seen as the screen play against the prevent defense. This is an obvious observation as these are basically opposite play/coverage combinations, but it makes us hopeful that this data will work well in our eventual model. Next we similarly wanted to look at how the number of defenders looks when compared to the averages yards gained by each route type.

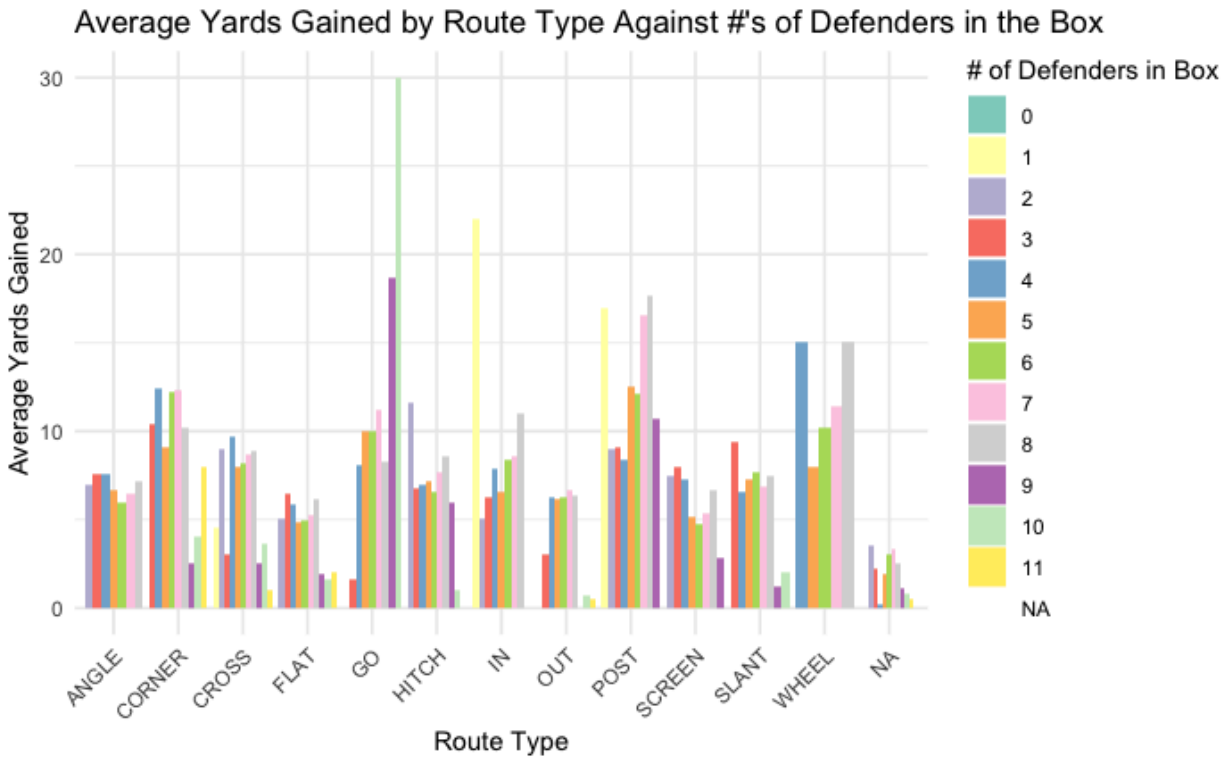


Figure 2

We can immediately see some strong correlations between deeper routes being more effective against a box that is stacked with more defenders. For example, if we look into the average yards gained on plays using a "GO" route, we can see that the greatest average yardage gain is when there are 10 defenders in the box. This is a very basic idea of football, such that with many defenders close to the line, a receiver running fast down the field will have a higher likelihood of catching a deep ball. The fact that the data shows this however, is again, a promising sign. Next we will take a closer look into the average Expected Points Added for each route type grouped by coverages.

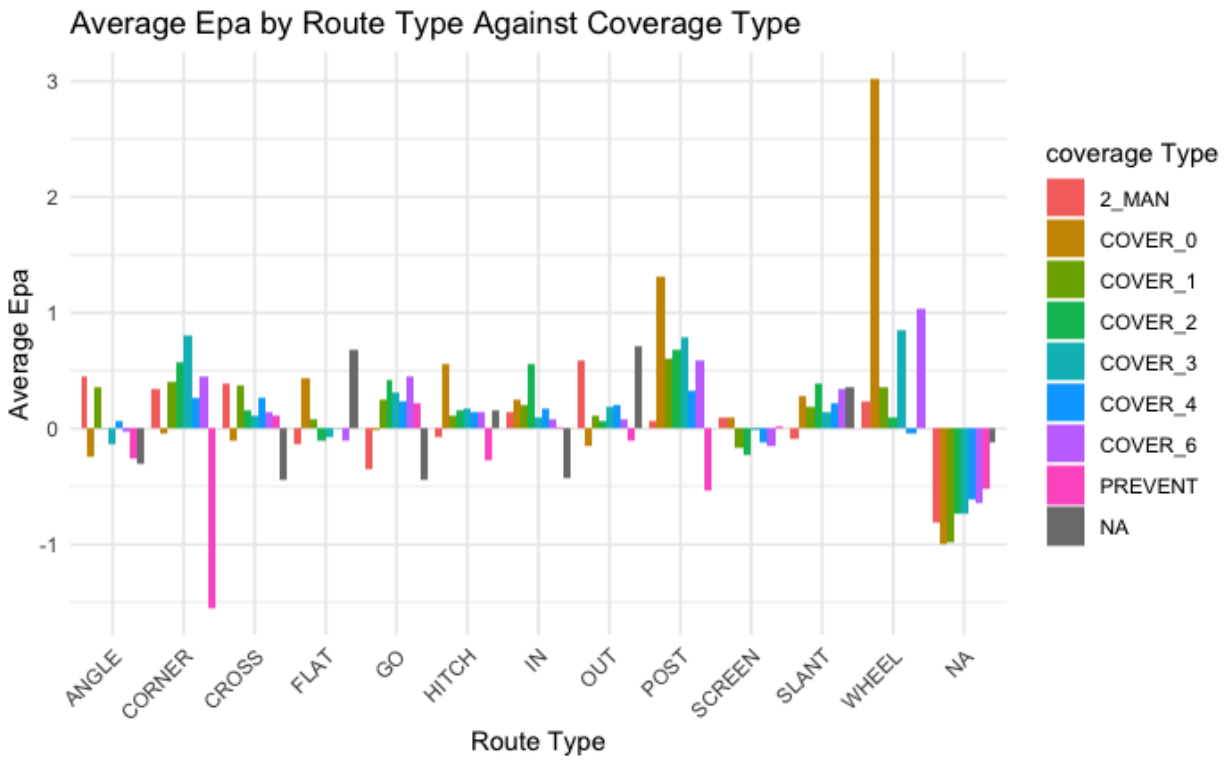


Figure 3

If we look at the corner routes, we are shown an interesting negative epa when facing a prevent defense. This means that the expected points added is actually negative for a corner route against a prevent defense according to our dataset. Basically this means that there is a higher likelihood of something going wrong when throwing this type of route against the prevent defense. This is very interesting and will prove to be important later. Lastly, we will look into a basic plot which shows the distribution of route types in our dataset from the 2023 NFL season.

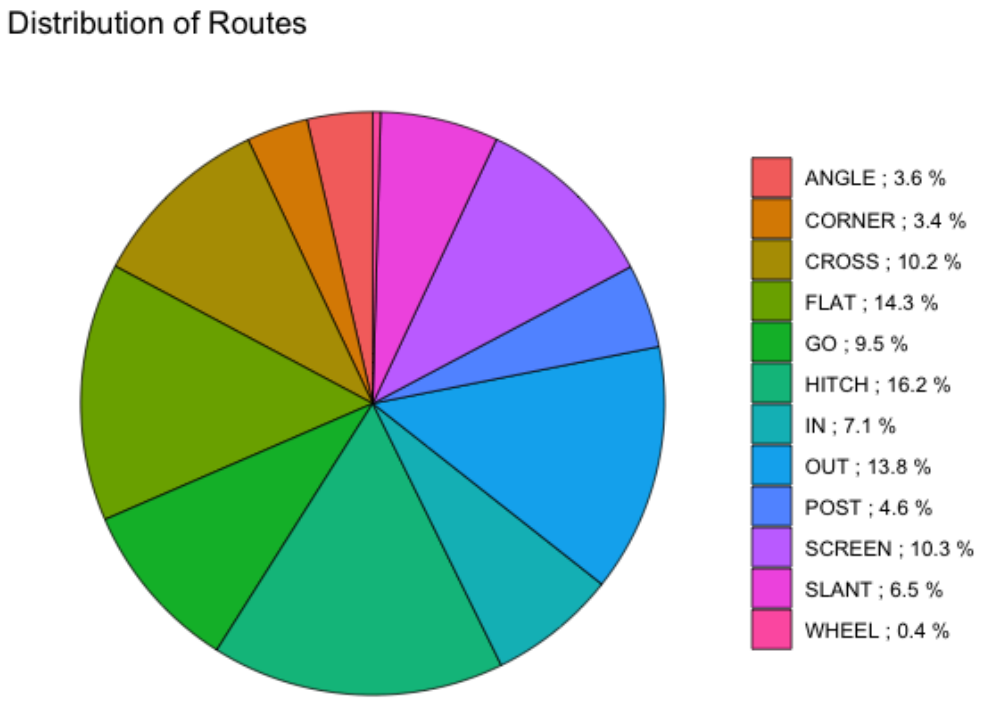


Figure 4

Exploratory Plotting for RQ2

For question 2, we need to focus on what defenses work against certain defensive coverages. Our first graph, *Figure 5*, starts us out by demonstrating what basic coverage types are ideal against individual formations. While this isn't a super detailed look into the defense, it gives us a general preview of what coverages defensive coordinators are generally calling. We can then refine this even more by personnel, as seen in *Figure 6*. This breaks it down more into how many defensive backs, linebackers, and defensive lineman are in the formation. This is important because not every cover 4 or different type of coverage contains the same personnel. This demonstrates the primary defensive personnel or formations that are being brought out to counteract certain offensive formations.

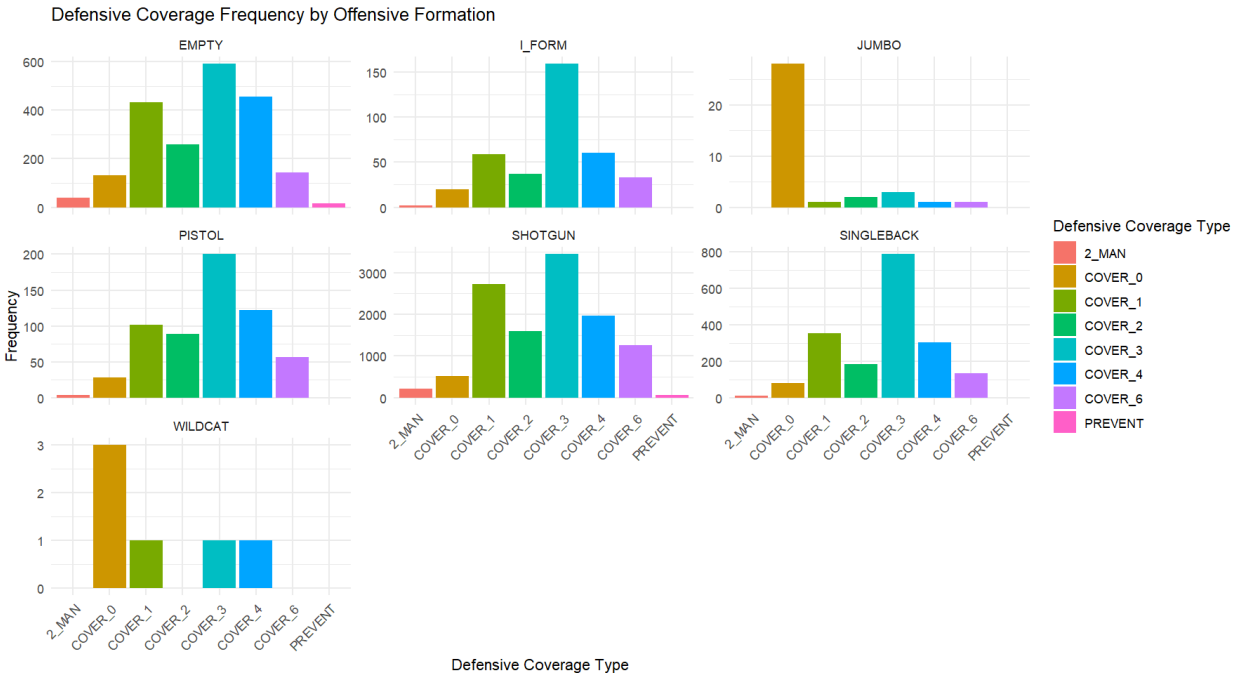

Defensive Coverage Frequency by Offensive Formation
EMPTY
I_FORM
JUMBO
PISTOL
SHOTGUN
SINGLEBACK
WILDCAT
Frequency
Defensive Coverage Type
2_MAN
COVER_0
COVER_1
COVER_2
COVER_3
COVER_4
COVER_6
PREVENT

Figure 5

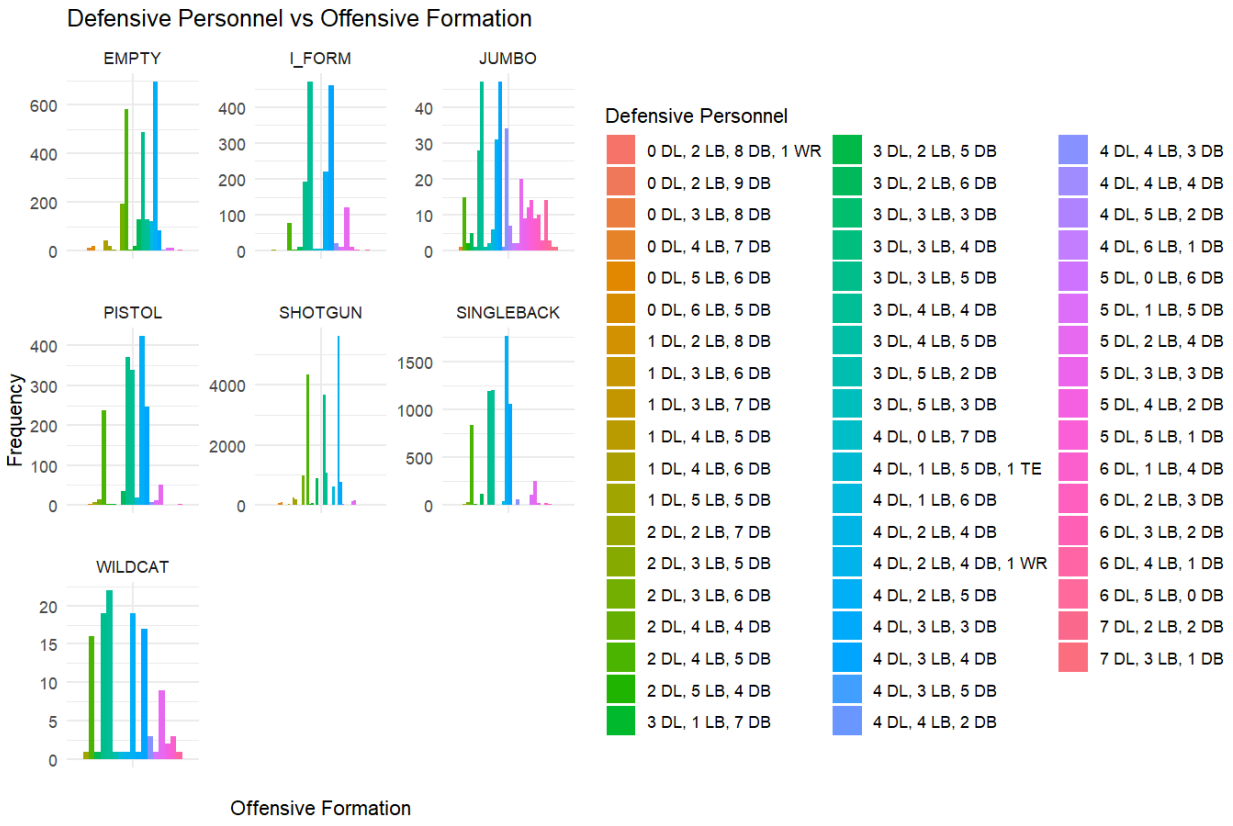


Figure 6

A heatmap, like demonstrated in *Figure 7,* can give us an even more accurate look into the prominent defensive personnel that are being utilized against certain formations. Our example looks into the pistol formation and the number of defenders in the box compared to the number of pass rushers. Which is important to show if the defense is showing a blitz, and then if they are actually blitzing or sending a less aggressive pass rush. We can get a general idea of the distributions of how many defenders we see in the box based on formation by looking at *Figure 8.* It also demonstrates the variance in selection that can be seen in the dataset, as in our example the shotgun formation has a very small distribution while the wildcat formation has a wide spread. This can allow us to compare the different approaches to different formations and train a model on the ideal selection.

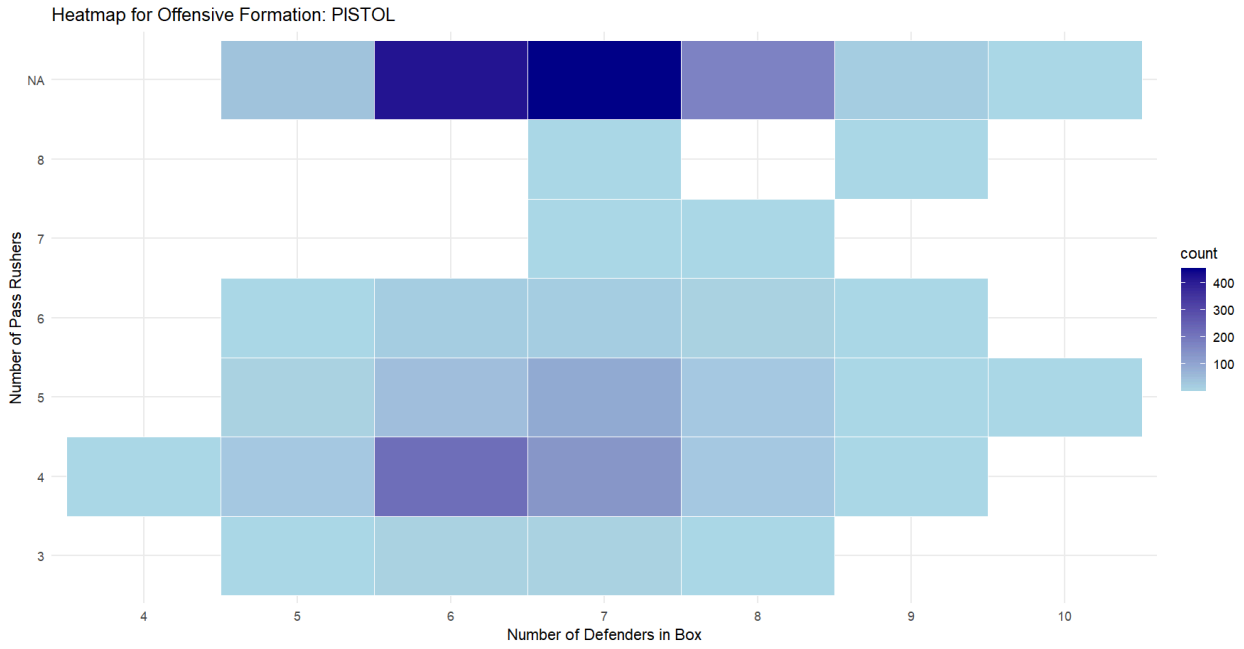


Figure 7

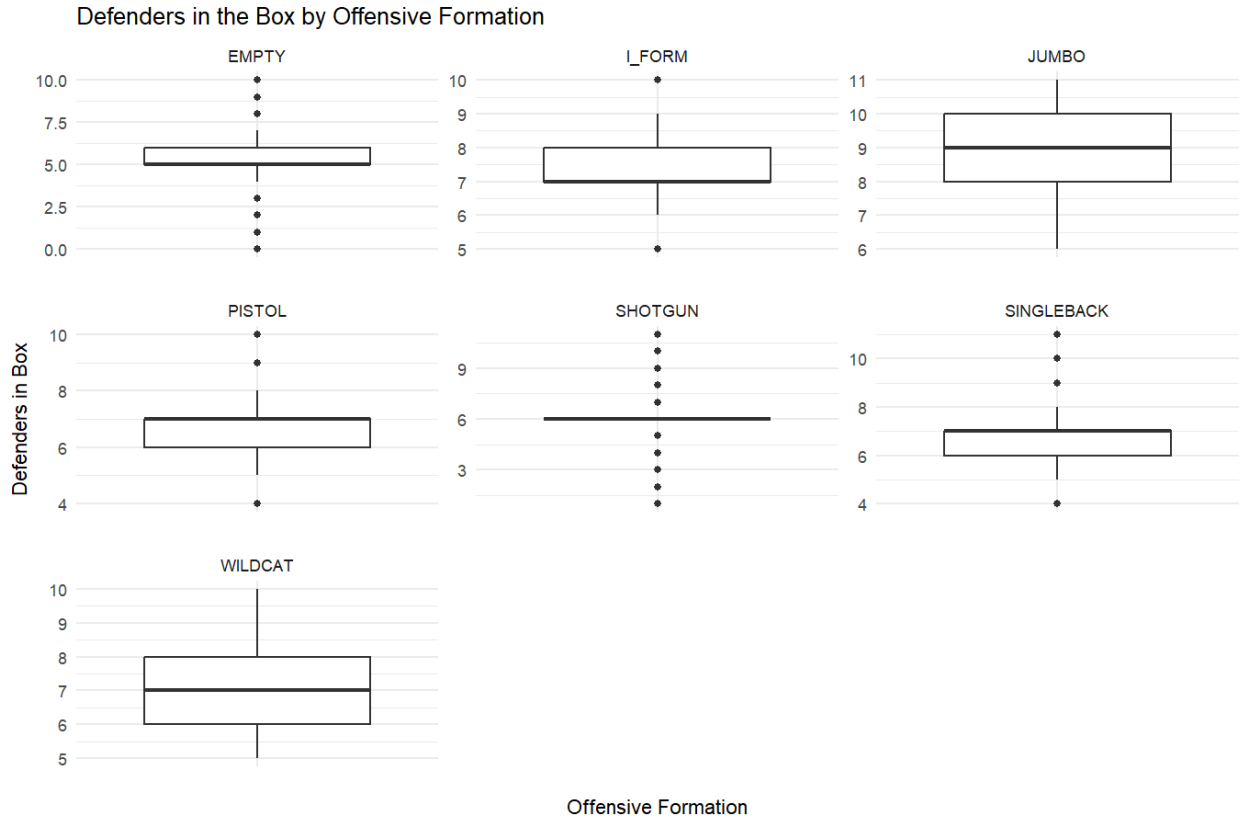


Figure 8

Exploratory Plotting for RQ3

For our final research question, we are attempting to track momentum. In order to do this, we will use multiple factors, like winning probability, EPA, and field position to demonstrate momentum. In *Figure 9*, the EPA by play for a game between the Chicago Bears and the Green Bay Packers is displayed to visualize the sway and trends of a game. Bigger plays will have a bigger impact on momentum and vice versa for negative plays. *Figure 10*, demonstrates the trend of field position and how impactful plays can occur from different areas of the field and the amount of impact that can have on the game. EPA shifts can demonstrate the severity of each play and how a big play from 75 yards out can sometimes be even more impactful than a big play from 20 yards out. Finally, *Figure 11* demonstrates the winning probability each team has and the trends that occur throughout an individual game. As seen in the example, there are many ebbs and flows that can be captured throughout each game and can be quantized into a metric that tracks momentum. Observe how the Bills winning probability was nearly 100% and then suddenly the Jets winning probability slowly creeps up until they are almost 100% and the Bills are 0%. These trends are important to show in a game that is often defined by momentum that can shift at a moment's notice.

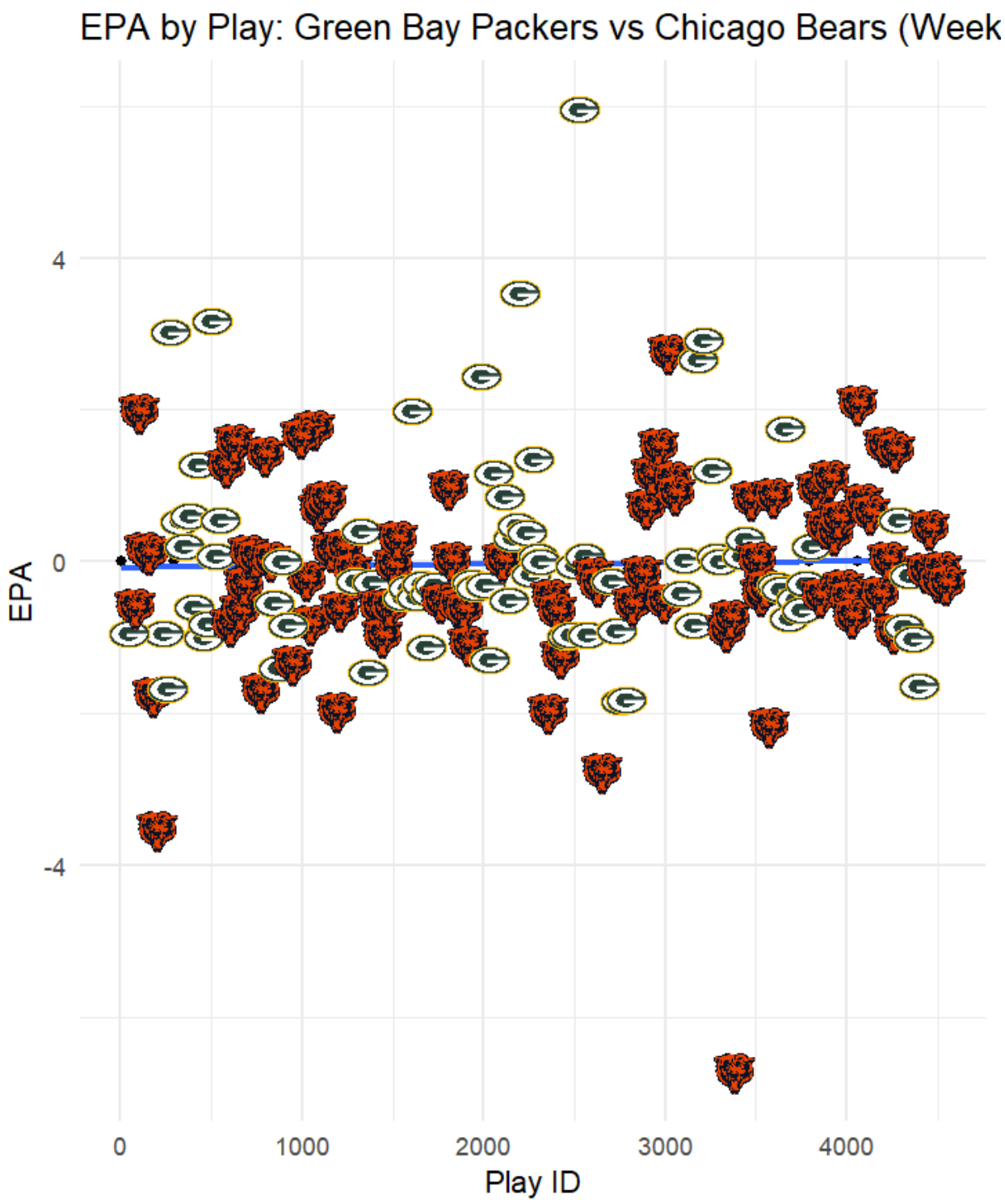

EPA by Play: Green Bay Packers vs Chicago Bears (Week
4
0
-4
EPA
0
1000
2000
3000
4000
Play ID

Figure 9

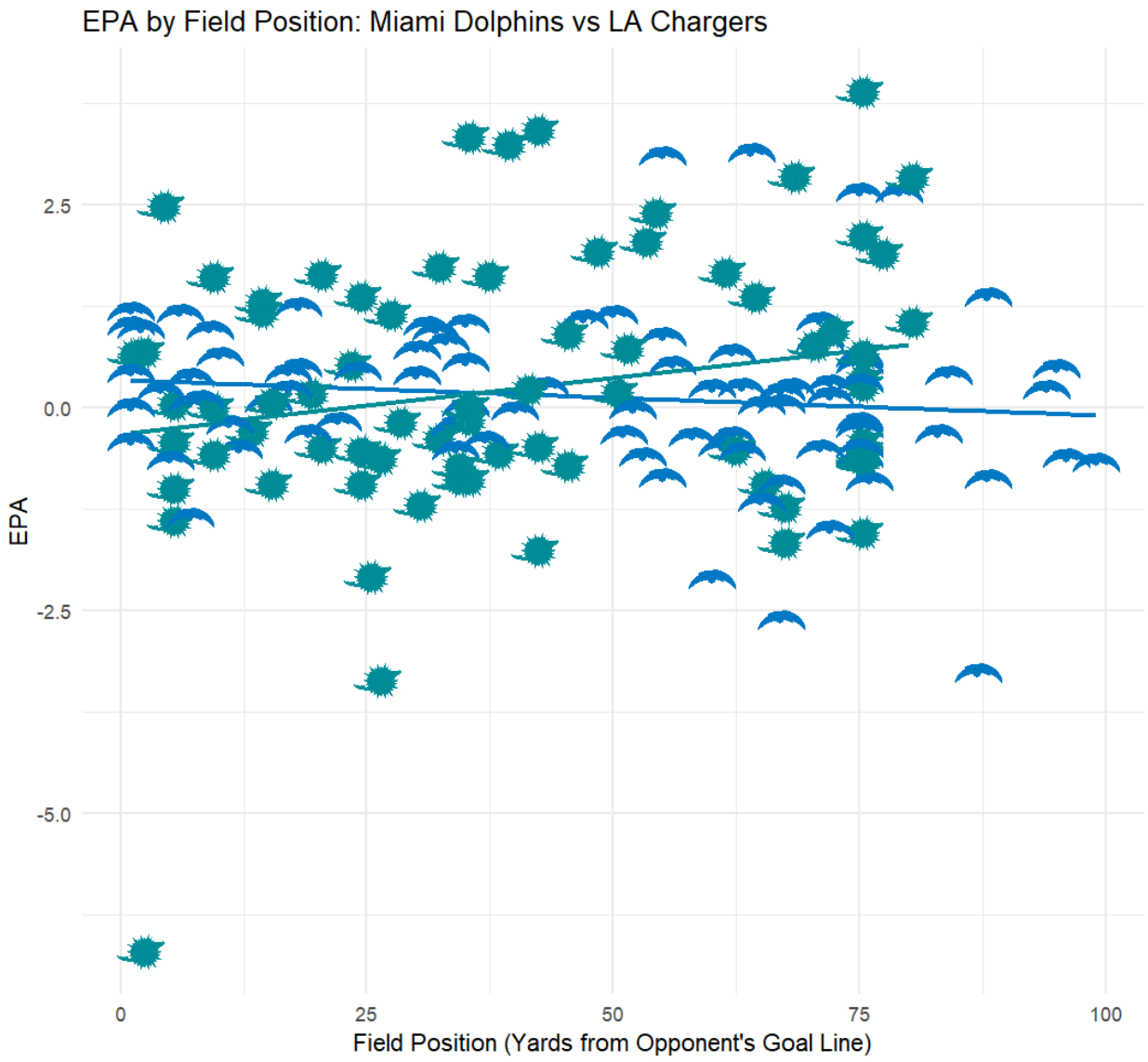

EPA by Field Position: Miami Dolphins vs LA Chargers
2.5
0.0
-2.5
-5.0
EPA
0
25
50
75
100
Field Position (Yards from Opponent's Goal Line)

Figure 10

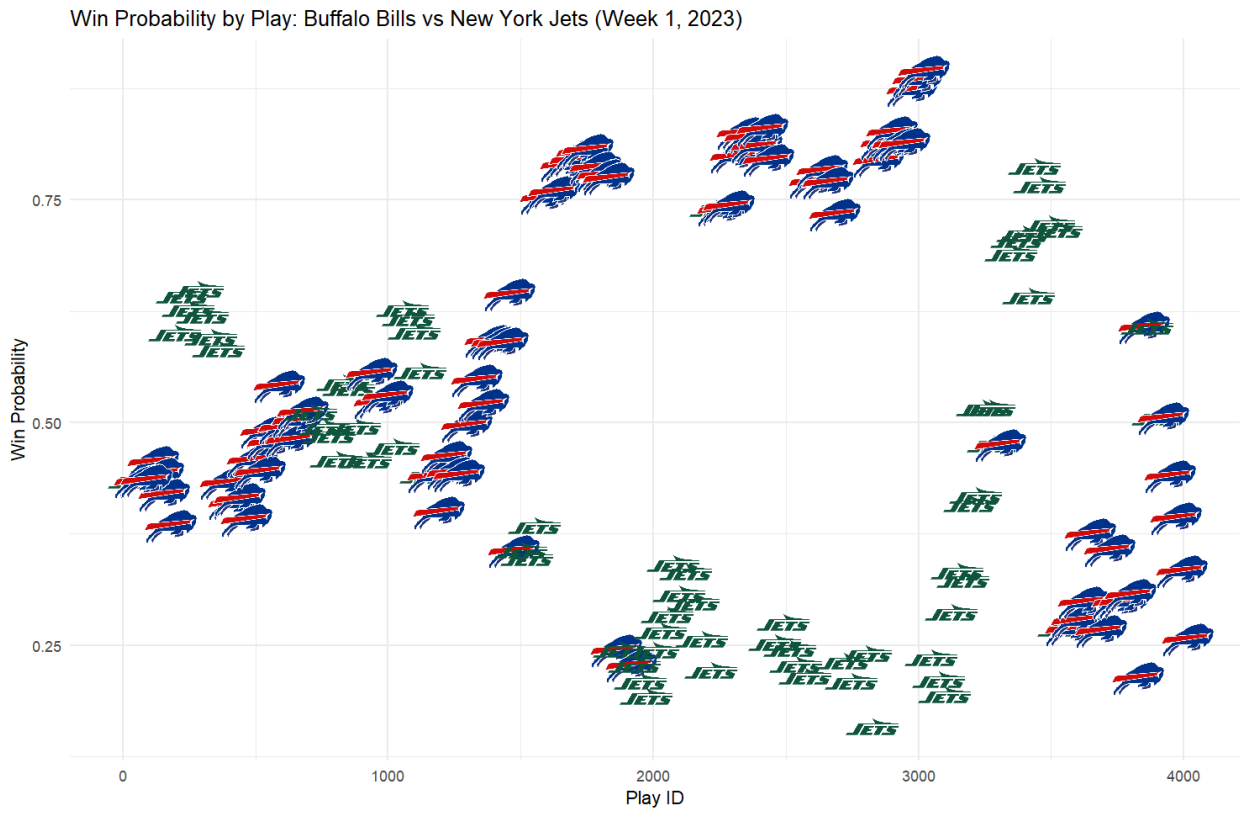


Figure 11

Methodology

Before diving straight into the creation of our three models, we first need to preprocess our data and make sure that not only is the data clean and free from incorrect or empty observations, but also that the data is in a format that will be easily accessible. This will make things much easier when we begin to split our data, and finally create and tune our models. In this document we will describe this process and all of the different aspects that go into it.

RQ1: What routes are ideal against certain defensive coverages?

For RQ1, we first needed to narrow down our original list of variables and create a subset using the following variables: **route**, **yards_gained**, **defenders_in_box**, **defense_coverage_type**, **pass_length**, **pass_location**, **epa**. This subset has dimensions of 16,000 rows and 7 columns after omitting 20 NA observations. This subset was obtained by filtering on the play_type variable being equal to pass and the route variable != NA. This will give us exactly what we need to create our model. For this question, our model will be a KNN due to its impressive performance when predicting multi-class categorical variables, as well as its sensitivity to class distribution. K-fold cross-validation will be used to optimally tune our model by providing a more reliable estimate of its performance, as it involves partitioning the dataset into *k* subsets and training the model across these different folds. This process helps in reducing overfitting, ensures that every data point gets to be in both the training and validation sets, and ultimately aids in selecting hyperparameters that yield the best generalization to unseen data. Certain categorical variables will need to be encoded to prepare the data for the KNN model. The defense_coverage_type and pass_location variables will require one-hot encoding, as they are non-ordinal categorical variables. Additionally, pass_length, which has values "short" and "long," will be handled with binary encoding since it represents an ordinal relationship. These encoding steps ensure that categorical data is properly transformed into a numerical format that the model can interpret. Other variables like yards_gained, defenders_in_box, and epa are continuous and will only need scaling rather than encoding. Using confusion matrices in addition to accuracy will provide deeper insights into the model's performance. This will account for the difference in true positives and false positives, making it a great choice for multi class categorical questions such as these.

RQ2:

For **RQ2**, we aimed to determine the ideal defensive coverages against specific offensive formations by leveraging both **Random Forest** and **Gradient Boosting** models. Our focus was on predicting which defensive coverage types would be most effective in countering particular offensive formations. The key variables used for this analysis included **defense_coverage_type** (target variable), **offense_formation**, **number_of_pass_rushers**, **sack**, **interception**, **yards_gained**, and **yardline_100**. These variables were selected for their relevance in capturing both the structure of the offense and key defensive outcomes, such as sacks and interceptions. After cleaning the data by removing any rows with missing values, the dataset was split into training and test sets using an 80-20 split, ensuring that the model's performance could be validated reliably.

For the **Random Forest** model, we employed a grid search to tune the **mtry** hyperparameter, which controls the number of variables randomly sampled at each tree split. We trained the model using 500 decision trees to enhance performance and applied 5-fold cross-validation to prevent overfitting. This method helped ensure the generalizability of the model by training and validating on different subsets of the data. After training, we used a confusion matrix to evaluate the model's predictions on the test set. The **variable importance plot** from the Random Forest model indicated that variables like **offense_formation**, **number_of_pass_rushers**, and **yards_gained** were the most influential in determining the predicted defensive coverage. These variables played a significant role in influencing the defensive strategy selected for any given play.

In addition to the Random Forest model, we implemented a **Gradient Boosting Model (GBM)** to further explore the relationships between offensive formations and defensive coverages. GBM, known for combining weak learners (decision trees) into a stronger predictive model, was employed with careful tuning of hyperparameters such as **learning rate**, **number of trees**, and **tree depth**. The grid search allowed us to refine these parameters, ensuring the model improved iteratively by correcting the errors of previous trees. Similar to the Random Forest model, 5-fold cross-validation was applied to maintain model robustness and avoid overfitting. The Gradient Boosting model, while effective in predicting common defensive coverage types, struggled with rarer classes like **COVER_6**, likely due to the class imbalance in the dataset.

By using both models, we were able to identify which variables and combinations were most critical in predicting effective defensive coverage types against offensive formations. This comprehensive approach offered insights into how defensive strategies are influenced by different offensive setups, contributing valuable information for game strategy and performance analysis.

RQ3: How can we create a metric to track the momentum of a given game?

For RQ3, we first needed to narrow down our variables to those most relevant for tracking momentum. We created a subset using the following: score_differential, td_prob, fg_prob, epa, wpa, yards_gained, drive_ened_with_score, turnover, total_home_score, total_away_score, and yardline_100. This subset resulted in approximately 20,000 rows and 11 columns after filtering out any NA values. The key to answering this question is to develop a momentum score, calculated by combining changes in wpa, epa, and scoring-related variables. This score will quantify momentum shifts by reflecting how each play influences the game's flow, factoring in the critical impact of field position. To create the momentum metric, we will combine changes in

wpa and epa with scoring-related variables such as whether a drive ended with a score, a turnover occurred, or how close the team is to the line (using yardline_100). We will explore multiple regression models to predict this momentum score, starting with a linear regression model and then moving on to more complex models, such as random forest regression and gradient boosting machines (GBM), which can capture nonlinear relationships. These advanced models will help account for significant game events, such as turnovers or big plays, as well as the impact of field position on momentum. Data preparation is essential for model performance. Continuous variables like yards_gained, epa, and yardline_100 will be scaled, while binary variables like turnover and drive_ended_with_score will be encoded. The dataset will split into an 80/20 training and test set to evaluate model accuracy. We will also implement K-fold cross-validation to tune hyperparameters and reduce overfitting, ensuring that the model generalizes well to unseen data. Evaluation metrics such as RMSE, MAE, and R-squared will help us determine the accuracy of our momentum predictions and identify which model performs best. This methodology will allow us to create a robust metric that tracks the ebb and flow of game momentum, providing insight into how specific plays and field position shift the dynamics of a game.

Data Visualizations

Model Summary and Analysis:

RQ1:

For RQ1, we begin by refining our initial list of variables to create a focused subset that includes **route, air_yards, yards_gained, defenders_in_box, number_of_pass_rushers,**

time_to_throw, defense_coverage_type, pass_length, pass_location, epa, and comp_air_epa. This refined subset consists of 16,000 rows and 7 columns after excluding 20 NA observations. The subset was generated by filtering for plays where the play_type variable is equal to "pass" and the route variable is not NA. Additionally, we applied a filtering process that ensures only successful plays are included by selecting observations based on a required yards gained to yards to first down ratio. For our analysis, we employ a Random Forest model, known for its robustness in handling multi-class categorical variables. To optimize the model, we utilized grid search to determine the best number of folds for k-fold cross-validation, allowing us to fine-tune the model parameters effectively. This method helps reduce overfitting, ensures that every data point is included in both training and validation sets, and aids in selecting hyperparameters that improve generalization to unseen data. To prepare the data for the Random Forest model, we need to encode certain categorical variables. The **defense_coverage_type and pass_location** variables will undergo one-hot encoding, while the **pass_length** variable, which has "short" and "long" values, will be treated with binary encoding due to its ordinal nature. Continuous variables like **yards_gained, air_yards, time_to_throw, comp_air_epa,** and **epa** will require scaling rather than encoding. In addition to accuracy metrics, we will employ confusion matrices as our visualizations to gain deeper insights into the model's performance, allowing us to account for the disparities between true positives and false positives, making this approach especially valuable for multi-class categorical analyses such as this one.

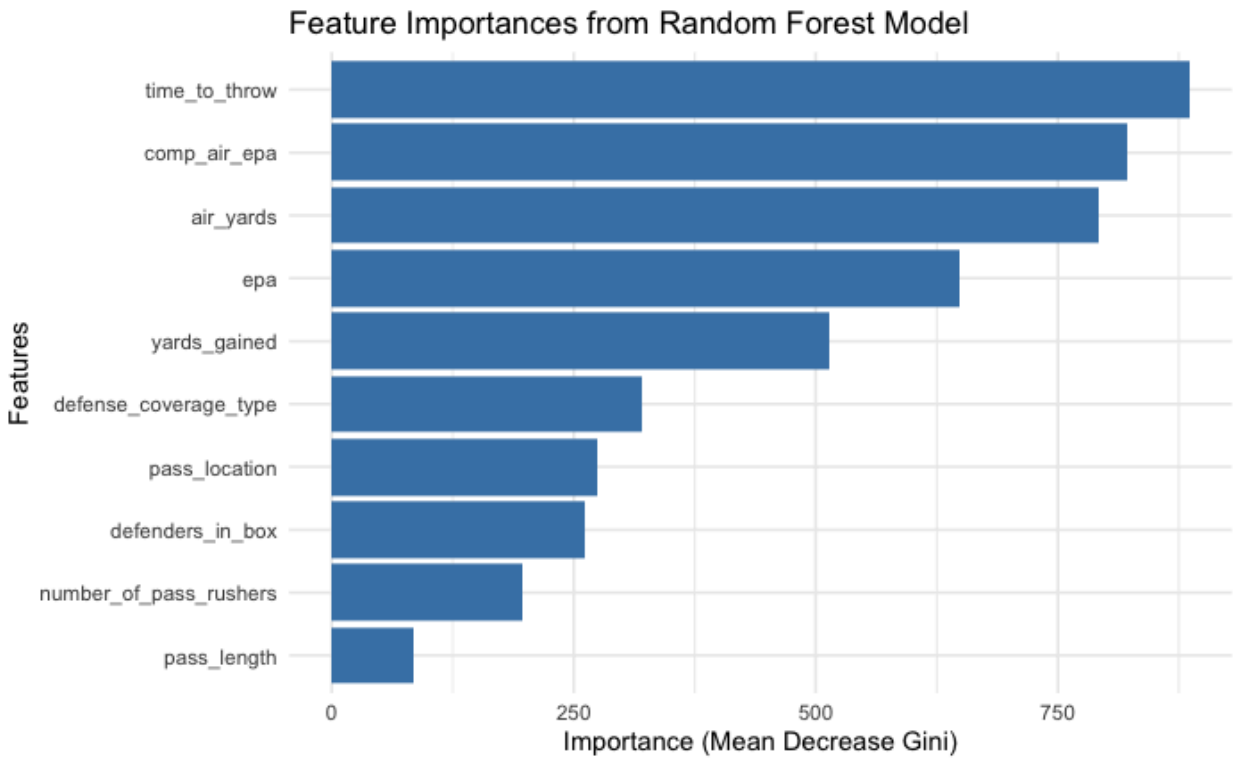


Figure 12

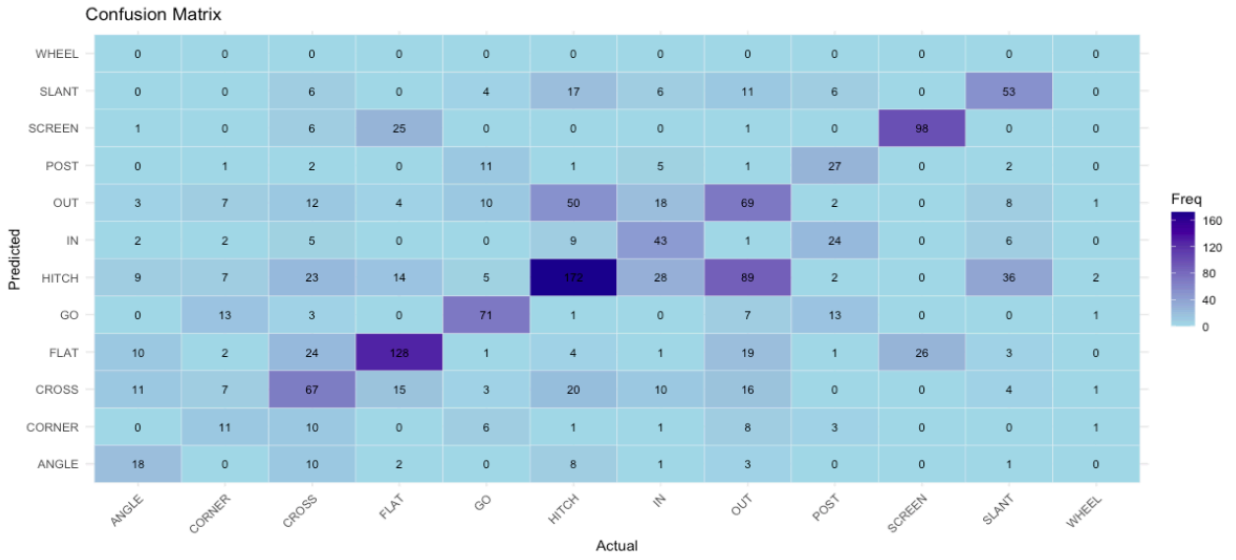


Figure 13

```
Overall Statistics

               Accuracy : 0.5003
                 95% CI : (0.4748, 0.5258)
    No Information Rate : 0.187
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4317
```

Figure 14

Our findings indicate an accuracy of around 50% for the Random Forest model, which reflects the inherent complexity and randomness associated with football as a sport. In the context of football, this level of accuracy can still provide valuable insights for various applications. Firstly, football is characterized by numerous variables that can influence the outcome of each play, such as player decisions, defensive strategies, weather conditions, and even psychological factors. The random nature of these elements means that while predictive models can identify patterns and trends, they cannot account for every variable that may impact play outcomes. As it is, the model seems to produce the majority of false positives on routes that are extremely similar such as an out route and a hitch, which are thrown at roughly the same point. Additionally, even with 50% accuracy, the model can still identify certain defensive strategies or play formations that are statistically more effective under specific conditions, allowing coaches and analysts to make informed decisions based on data rather than intuition alone.

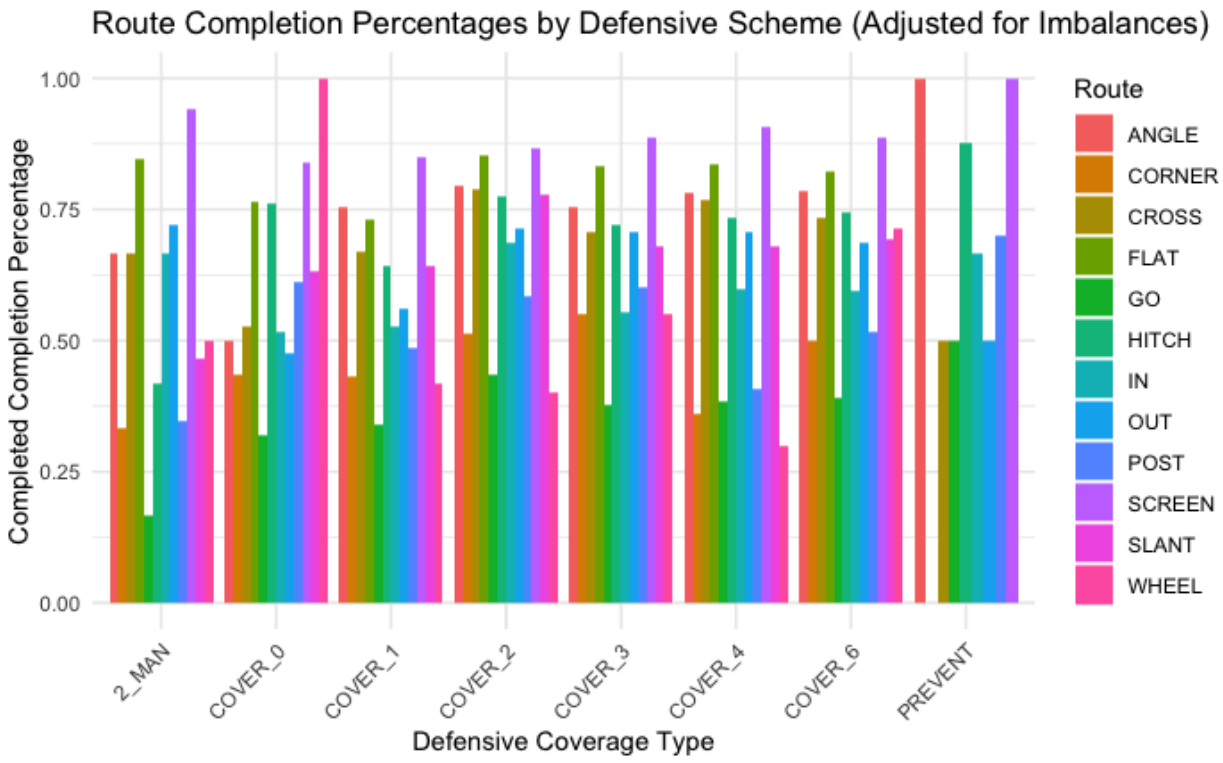


Figure 15

RQ2: For RQ2, we focus on predicting the optimal defensive coverage type based on a set of offensive indicators and contextual features. To create the initial subset, we selected variables including **offense_formation, yards_gained, yardline_100, offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers, and xyac_epa**, ensuring all records contain non-missing values for the critical variables. After excluding NA observations, we retain 13,000 rows and 8 columns for the analysis.

For this analysis, we employ a Gradient Boosting Model (GBM), which is well-suited for handling both categorical and continuous variables and is effective in capturing complex interactions between variables. We tune the GBM model using grid search over hyperparameters such as the number of trees, tree depth, learning rate, and minimum observations per node, which

are optimized through 3-fold cross-validation. This approach helps balance runtime efficiency with the model's ability to generalize well to unseen data.

In preparing the data for the model, categorical variables like offense_formation, offense_personnel, and defense_personnel are encoded as factors. Continuous variables like yards_gained, yardline_100, and xyac_epa are standardized, while defenders_in_box and number_of_pass_rushers are filled with the median values where necessary.

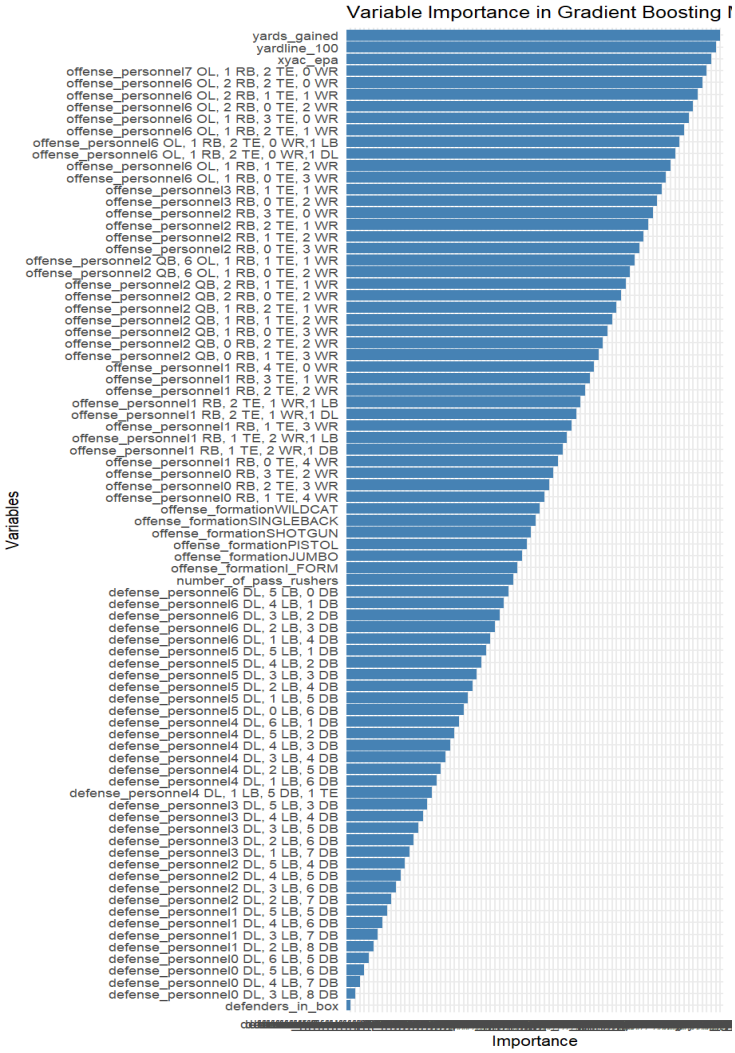


Figure 16

To evaluate model performance, we primarily rely on the ROC curve to assess how well the model distinguishes between different defensive coverages. The ROC curve allows us to measure the trade-off between true positives and false positives for each defensive coverage type, providing a clear indication of the model's precision and recall across classes. Additionally, we employ a variable importance plot, which highlights the most influential features in predicting defensive coverage. This plot is critical in understanding which offensive indicators (e.g., offense_formation, yarline_100, and yards_gained) have the most impact on the model's predictions.

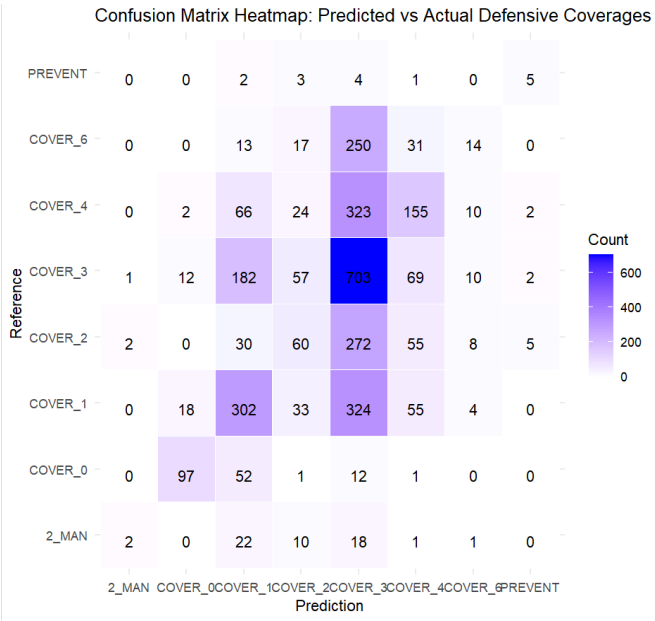


Figure 17

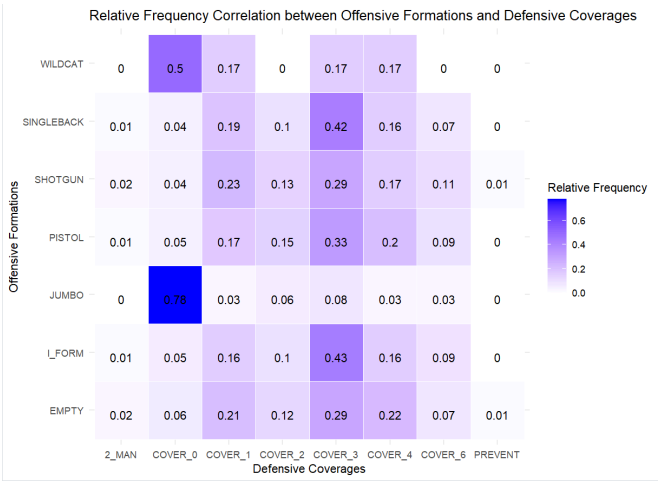


Figure 18

The visualizations displayed in the two heatmaps provide insights into the model's performance and the relationships between offensive formations and defensive coverages. The confusion matrix heatmap on the left compares the predicted and actual defensive coverages, illustrating how well the model predicts each type of coverage. The blue-shaded cells, representing higher frequencies, indicate where the model has successfully predicted defensive coverages, while

lighter cells show areas of misclassification. For instance, the model frequently predicts Cover 3 correctly, as indicated by the darker blue cell along the diagonal, but struggles more with Cover 0 and Cover 1, where predictions are spread across multiple coverage types. This confusion matrix helps us understand the model's accuracy across different coverage types and highlights where the model is performing well and where it may need improvement.

The relative frequency correlation heatmap on the right explores the relationship between offensive formations and defensive coverages. It displays the proportion of times each defensive coverage is used in response to a given offensive formation. For example, Jumbo is strongly associated with Cover 0 (78% of the time), while Singleback and Shotgun formations are more evenly distributed across multiple coverages. This visualization helps us answer the research question by showing the defensive coverages most frequently used for each offensive formation, identifying trends in strategic responses. These insights allow us to better understand which defensive coverages tend to be favored against specific offensive setups, which can aid in predicting the optimal defensive strategy based on historical data. Both visualizations are critical in evaluating the model’s effectiveness and understanding the relationships between offensive formations and defensive coverages, supporting the goal of predicting optimal defensive responses in football.

```
> print(confusion_matrix)
Confusion Matrix and Statistics

          Reference
Prediction 2_MAN COVER_0 COVER_1 COVER_2 COVER_3 COVER_4
   2_MAN       1       0       1       2       2       0
   COVER_0     0      99      22       1      13       2
   COVER_1    22      50     293      34     174      67
   COVER_2    15       1      38      60      68      25
   COVER_3    14      12     318     265     682     307
   COVER_4     1       1      57      59      78     165
   COVER_6     1       0       7       8      17      14
   PREVENT     0       0       0       3       2       2
          Reference
Prediction COVER_6 PREVENT
   2_MAN         1       1
   COVER_0       0       0
   COVER_1      21       2
   COVER_2      23       4
   COVER_3     225       3
   COVER_4      37       1
   COVER_6      18       0
   PREVENT       0       4

Overall Statistics

               Accuracy : 0.3955
                 95% CI : (0.3788, 0.4123)
    No Information Rate : 0.3099
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1975

 Mcnemar's Test P-Value : NA

Statistics by Class:
```

Figure 19

	Class: 2_MAN	Class: COVER_0
Sensitivity	0.0185185	0.60736
Specificity	0.9978717	0.98805
Pos Pred Value	0.1250000	0.72263
Neg Pred Value	0.9841079	0.98004
Prevalence	0.0161532	0.04876
Detection Rate	0.0002991	0.02961
Detection Prevalence	0.0023931	0.04098
Balanced Accuracy	0.5081951	0.79771
	Class: COVER_1	Class: COVER_2
Sensitivity	0.39810	0.13889
Specificity	0.85807	0.94023
Pos Pred Value	0.44193	0.25641
Neg Pred Value	0.83470	0.88035
Prevalence	0.22016	0.12923
Detection Rate	0.08765	0.01795
Detection Prevalence	0.19832	0.07000
Balanced Accuracy	0.62809	0.53956
	Class: COVER_3	Class: COVER_4
Sensitivity	0.6583	0.28351
Specificity	0.5041	0.91525
Pos Pred Value	0.3735	0.41353
Neg Pred Value	0.7666	0.85836
Prevalence	0.3099	0.17410
Detection Rate	0.2040	0.04936
Detection Prevalence	0.5462	0.11935
Balanced Accuracy	0.5812	0.59938
	Class: COVER_6	Class: PREVENT
Sensitivity	0.055385	0.266667
Specificity	0.984427	0.997897
Pos Pred Value	0.276923	0.363636
Neg Pred Value	0.906345	0.996699
Prevalence	0.097218	0.004487
Detection Rate	0.005384	0.001197
Detection Prevalence	0.019444	0.003290
Balanced Accuracy	0.519906	0.632282

Figure 20

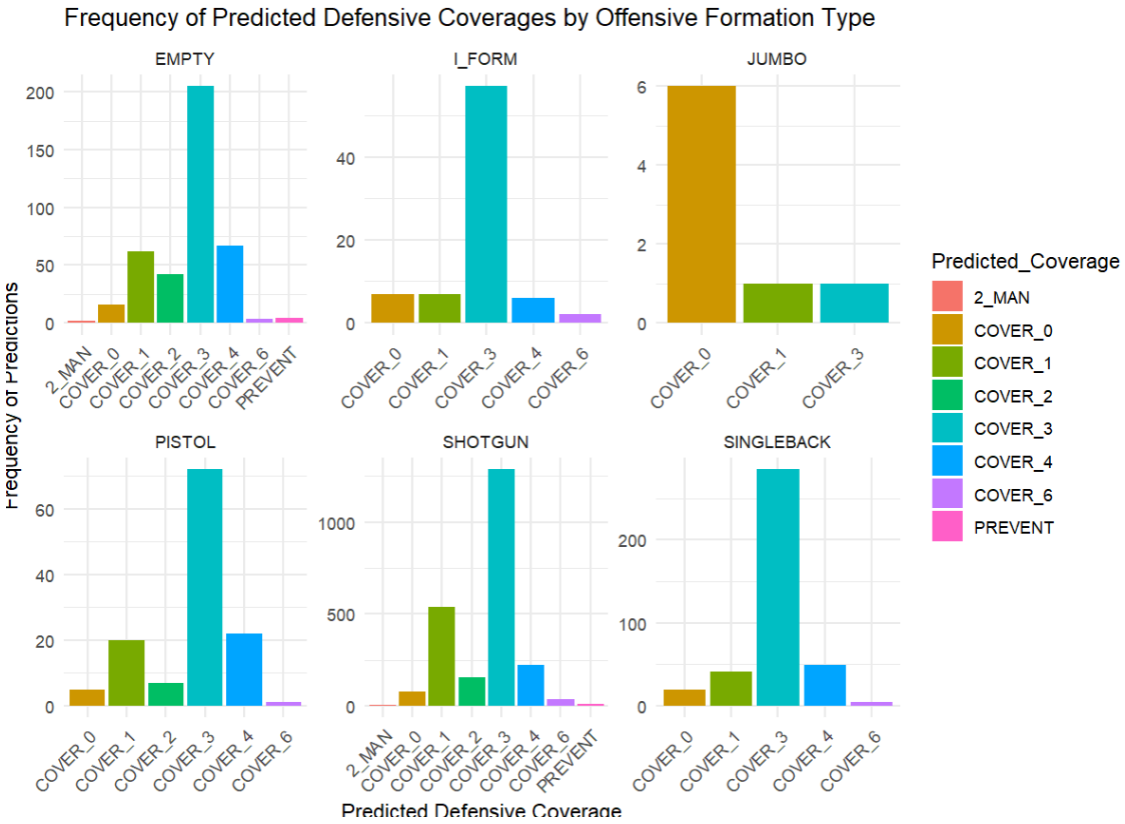


Figure 21

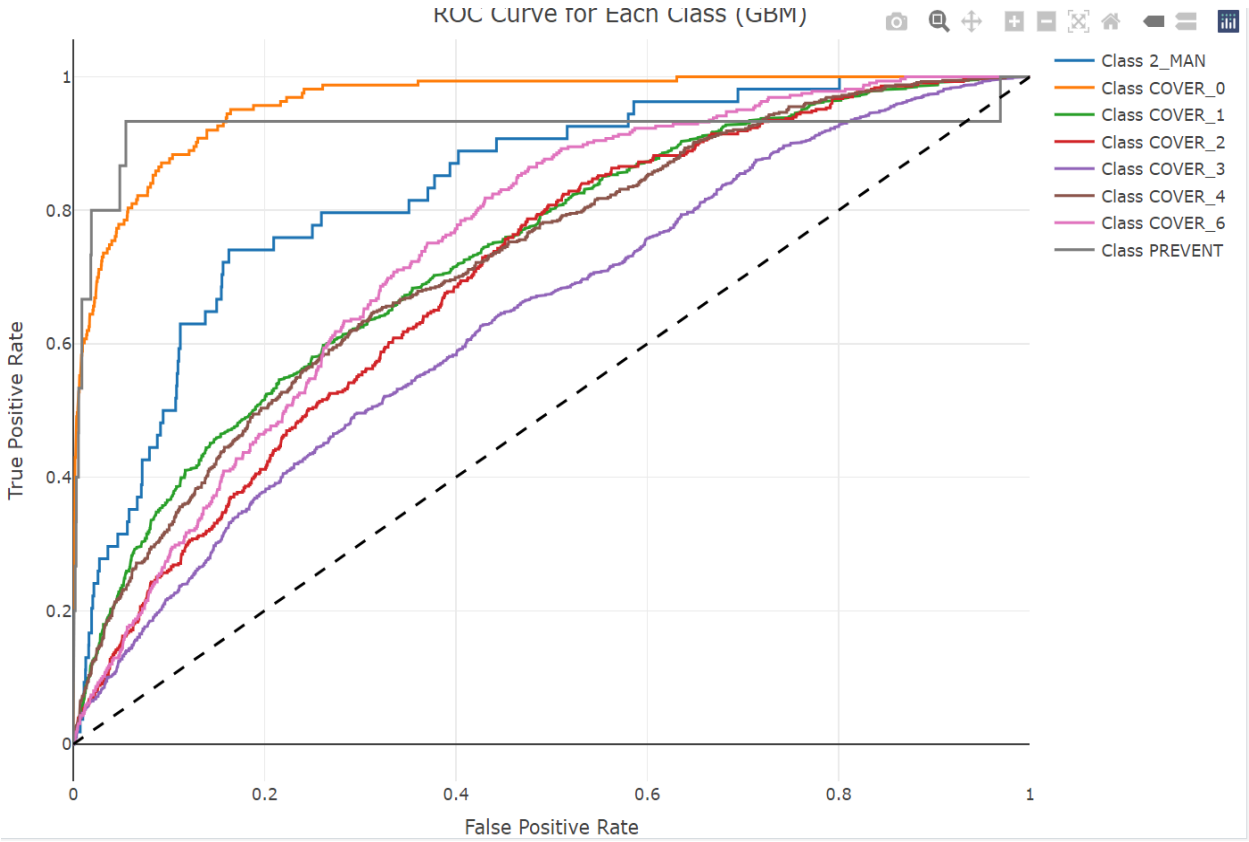


Figure 22

Our findings show an accuracy of approximately 40% for the GBM model. While this may seem modest, it reflects the complexity of predicting football plays, where outcomes are influenced by numerous dynamic and often unpredictable factors. In the context of football analytics, even models with moderate accuracy can provide valuable insights. Defensive strategies are multifaceted, and predicting their success based solely on offensive indicators presents a challenging task. Despite this, the GBM model allows us to identify key trends and patterns, helping coaches and analysts make more data-driven decisions. The inclusion of features like offense_personnel and xyac_epa ensures that the model captures essential aspects of play success, and the use of the ROC curve and variable importance provides valuable insights into the model's performance and the relative influence of key features.

RQ3: For RQ3, we aim to evaluate the predictive accuracy of momentum scores using two machine learning models: Random Forest and Gradient Boosting (GBM). We start by constructing a momentum score that incorporates weighted factors such as changes in WPA (Win Probability Added), EPA (Expected Points Added), scoring events, and turnover events. This score reflects the dynamic shifts in momentum throughout a game. The dataset includes variables such as **score_differential, yards_gained, wpa, and epa**, with missing values for numeric variables handled via median imputation. After preprocessing, the data subset contains key features relevant to momentum prediction.

We first apply a Random Forest model, tuning it with cross-validation to determine the optimal number of predictors sampled (mtry). The R-squared vs. number of predictors sampled plot illustrates the model's high accuracy in fitting the data, indicating that the model effectively captures the relationship between the features and the momentum score. Additionally, the RMSE vs. number of predictors sampled plot shows that increasing the number of predictors improves the model's performance by reducing prediction error. The final Random Forest model demonstrates low RMSE, indicating a strong alignment between predicted and actual momentum scores.

```
Random Forest

31735 samples
   15 predictor

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 21157, 21157, 21156
Resampling results across tuning parameters:

  mtry  RMSE        Rsquared   MAE
  2     0.15050313  0.9630561  0.08145346
  4     0.06407545  0.9919411  0.02477951
  6     0.03760715  0.9970887  0.01146363

RMSE was used to select the optimal model using
 the smallest value.
The final value used for the model was mtry = 6.
```

Figure 23

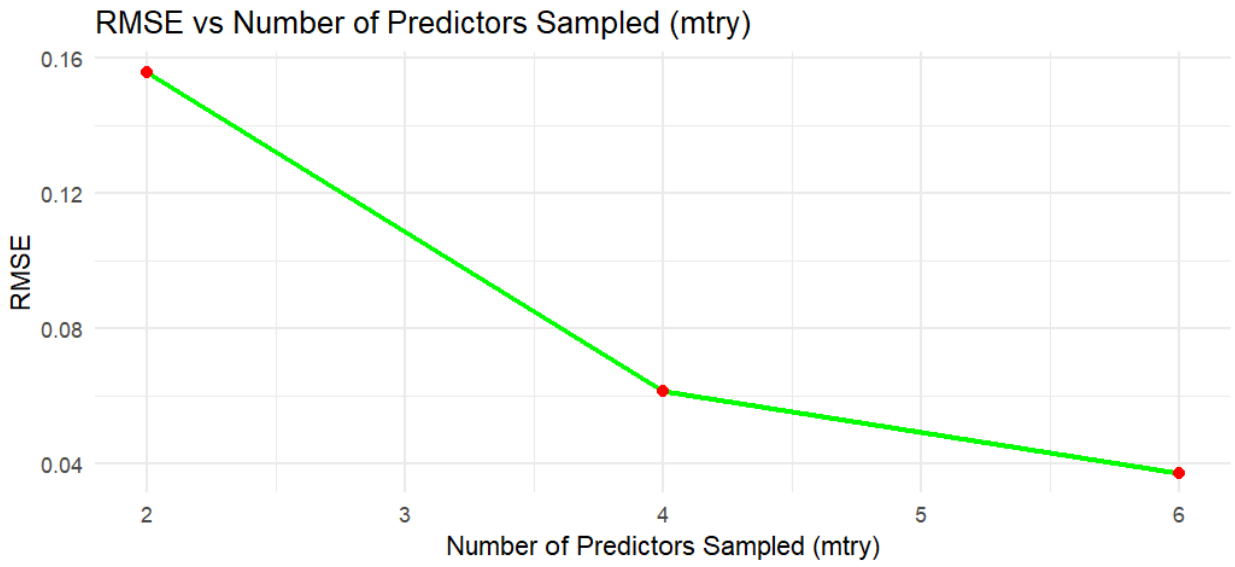


Figure 24

Next, we evaluate the performance of the Gradient Boosting Model (GBM). We tune the GBM model by adjusting the number of trees and tree depths, optimizing performance through cross-validation. The RMSE vs. number of trees plot demonstrates that as the number of trees increases, the model's accuracy improves, particularly with deeper trees. This indicates that the GBM model effectively captures complex interactions between the features, contributing to

accurate momentum predictions.

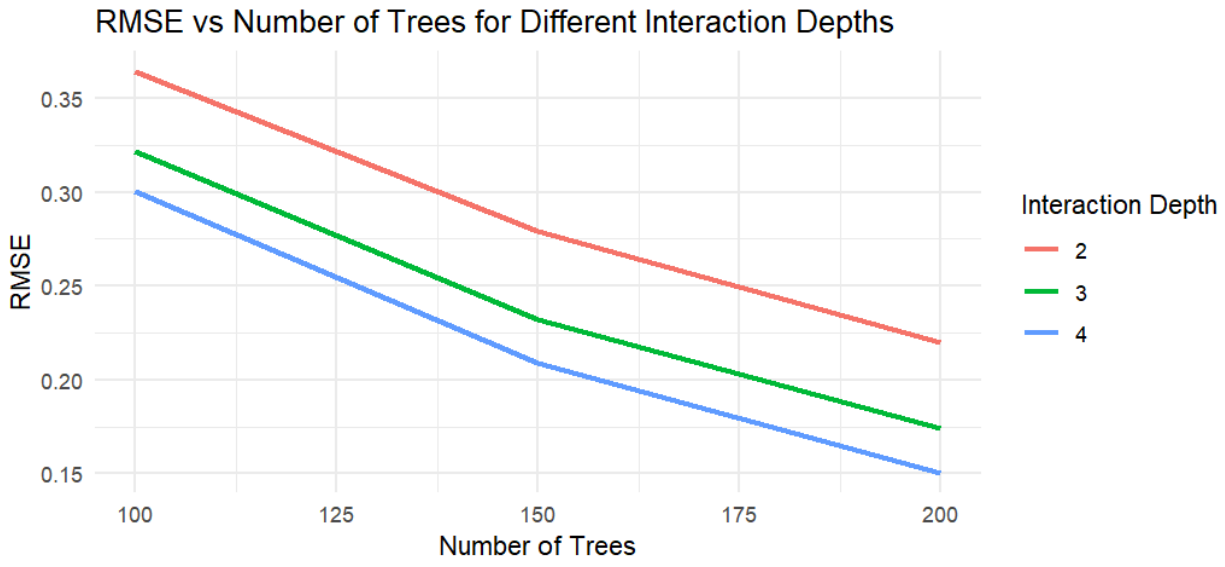


Figure 25

One of the key visualizations in this analysis is the predicted vs. actual momentum score plot for both the Random Forest and GBM models. The plot shows a strong correspondence between the predicted and actual values, with points closely following the reference line. This reflects the models' ability to accurately predict momentum scores. The analysis focuses on how well the models can capture the dynamic shifts in momentum using variables like delta_wpa, delta_epa, and scoring events, which play a central role in determining shifts in momentum during a game.

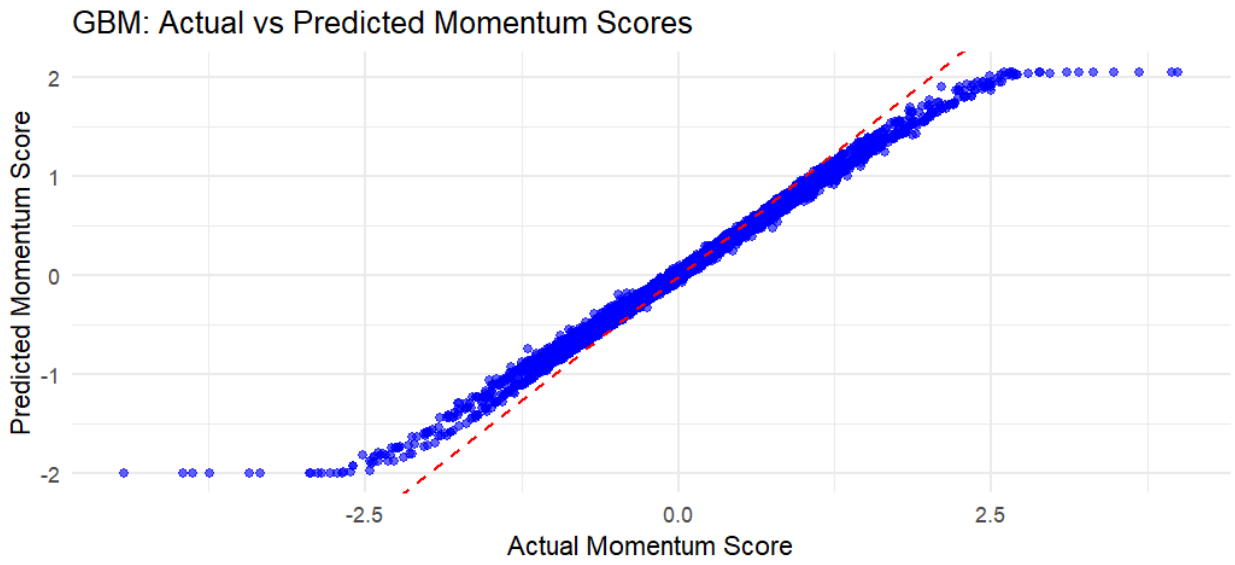


Figure 26

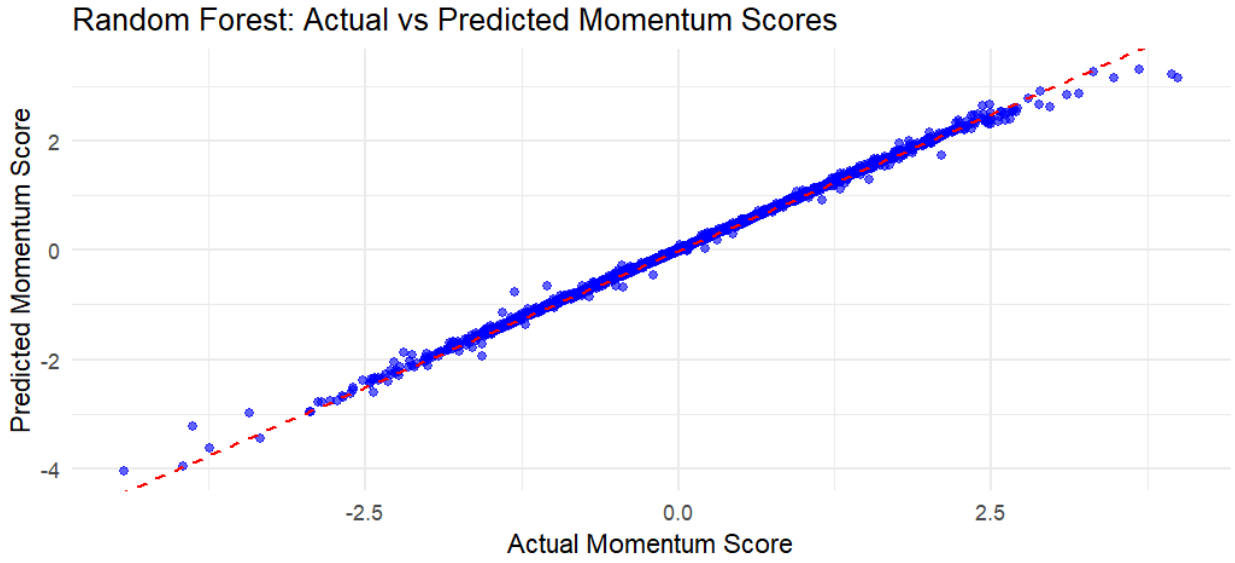


Figure 27

Overall, the analysis shows that both the Random Forest and GBM models perform well in predicting momentum scores, with high R-squared values and low RMSE. While both models

provide strong predictions, the Random Forest model slightly outperforms GBM in terms of accuracy, making it the preferred approach in this context. These findings demonstrate that machine learning models can effectively quantify momentum shifts in football, providing insights for analysts and coaches.

Ethical Recommendations

Data tracking and the use of Machine Learning is far from new to the world of primetime sports such as the NFL. As data tracking has become increasingly prevalent over the past few years with the tremendous increase of sports betting across the country, it is only natural that this ultra-precise data be used in a Machine Learning Context. However, the widespread adoption of these technologies also raises concerns about the ethical implications, including issues of fairness, privacy, and the potential for over-reliance on predictive models in a game as unpredictable as football.

For RQ1: The use of advanced ML models for the purpose of predicting route optimization could, in the wrong environment, create an uneven playing field. Teams with more resources to invest in cutting-edge technology could gain an unfair advantage, leading to disparities in performance and competition. Historical data, when used for event prediction in a dynamic and unpredictable environment like the National Football League, can create a false sense of security in decision-making. This is because past performance may not always accurately predict future outcomes, especially in a sport where randomness and unforeseen factors often play a significant role. It's important to make sure clients using this tool understand that it provides an analytically based estimate, not a guaranteed outcome. This helps teams

incorporate machine learning into their strategy while recognizing that it should not be the sole factor in decision-making.

For RQ2: Identifying optimal defensive coverages against offensive formations, there is a risk of embedding biases from historical data into predictive models, which may lead to the reinforcement of existing defensive strategies over exploring innovative ones. Historical data often reflects past tendencies and biases, and models trained on such data could favor traditional defensive setups that align with historical success rather than evolving tactics (fairmodels, 2021). This feedback loop may stifle innovation in defensive strategies, narrowing the range of defensive formations that teams employ. Ensuring fairness and innovation in defensive analytics requires regularly auditing and updating these models to avoid outdated biases, allowing teams to explore a changing array of defensive strategies.

For RQ3: Developing a metric to track game momentum, quantifying something as intangible as "momentum" introduces the risk of oversimplifying the complex game dynamics. While momentum metrics can provide valuable insights, they could also be misinterpreted by media and fans, leading to unfair criticism of players or teams based on perceived shifts in game momentum that are driven by algorithmic calculations rather than actual game context (MDPI, 2020). Additionally, momentum tracking could influence coaching decisions in ways that undermine situational judgment, as coaches may feel pressured to adjust strategies based on the model's interpretation of momentum rather than their own in-game assessments. Transparency about how momentum is quantified, as well as the limitations of these models, is essential to avoid over-reliance on algorithmic metrics in such a fluid and context-sensitive area of sports.

In addressing these ethical considerations, it becomes clear that applying predictive analytics in NFL play calling and momentum tracking requires a balance of fairness,

transparency, and respect for player autonomy. While Machine Learning is not a new practice for the NFL, its increasing integration into strategy development, player analytics, and game predictions highlights the need for teams to balance data-driven insights with human expertise, ensuring that technology enhances rather than dictates decision-making. Responsible analytics should enhance decision-making without restricting the flexibility of coaches and players to adapt in real time. Ensuring equitable access to analytics tools across all teams will also help preserve competitive fairness, preventing advantages for wealthier organizations. By updating models regularly and maintaining open communication about the purpose and limitations of these analytics, the NFL can support a data-informed environment that respects the sport's complexity and the role of human judgment.

Challenges:

For RQ2, determining the best set of variables was very challenging as there were a lot of defensive formation variables to select from. Not only are there the 8 standard formations but there is also a lot of variation in personnel that the model had to factor in. This was causing a slow runtime when determining variable importance and also when getting an output for the model. Additionally, the data had a large frequency in specific offensive formations, such as the shotgun formation, which made the data difficult to visualize and train accurately. Additionally, certain variables were showing a poor false positive value on the ROC curve, and removing these variables was found to be more difficult than initially thought, as the variables were worked within a pre-grouped set of data. Additionally, there were a decent amount of NA values throughout the dataset that required cleaning and repairing of the dataset in order to properly

train our model. These challenges all contributed to difficulty along the way to answering if we can develop a model that can predict optimal defensive coverages against offensive formations.

For RQ3, it was a challenge at the beginning mostly, due to the fact we had to develop an entirely new metric on a unit that is not extremely tangible, momentum. In order to do this, we had to determine feature importance and what led to increases in teams winning probability and trends that could be highlighted in the game. Ultimately, we generated a formula that valued EPA, yards gained, and other variables that generated a metric that could be used to quantify the surges of a football game. Another challenge is obviously we have no metric to compare our actual value to, but we have multiple metrics we can compare relatively to, such as changes in winning probability, as well as the actual score of the game. While this is a complex question to address, our metric produces a consistent and reliable representation of momentum which answers our research question.

Recommendations:

Based on our findings, we recommend several steps for NFL teams, data analysts, and future research to maximize the benefits of predictive analytics in football strategy. First, teams can enhance play-calling strategies by leveraging data insights that highlight specific route-coverages with high success rates. For instance, offensive coordinators could benefit from incorporating quick slants or deep posts when facing man-to-man or Cover 2 defenses. Integrating these insights into play-calling, particularly through automated data feeds connected to real-time analysis tools, would enable teams to adapt to the defense in-game.

Defensive play-calling could also improve by focusing on pattern recognition. Training defensive coordinators and players to recognize offensive formations that frequently correspond with specific plays can improve defensive stops. Automating such analysis through game footage could assist teams in identifying and preparing for opponents' tendencies. Additionally, the momentum-tracking metric we developed can provide critical insights into game dynamics and shifts. We recommend teams utilize this metric alongside real-time win probability and expected points added (EPA) to make informed decisions during large momentum swings. However, teams should be mindful of over-relying on the metric without considering situational factors, as misinterpreting game context could lead to less effective decisions. Future refinements, such as incorporating additional situational variables, could improve the model's predictive ability.

For continued improvement, future research could expand upon our models by incorporating more nuanced play and environmental variables, such as weather conditions or crowd noise, that might influence game outcomes. Additionally, increasing the focus on model explainability would be crucial; by assessing key variables affecting momentum or play success, coaches and analysts can gain a clearer understanding of the predictions and make confident decisions. Implementing these recommendations will strengthen NFL team's analytical and on-field adaptability, ultimately improving performance through data-driven strategies.

References:

Next-Gen Stats information:

https://operations.nfl.com/gameday/technology/nfl-next-gen-stats/

Superbowl Viewership information:

https://www.nielsen.com/news-center/2024/super-bowl-lviii-draws-123-7-million-average-viewers-largest-tv-audience-on-record/

NFL information:

https://www.nfl.com/

Stanford University. "Predictive Modeling in NFL Play Outcomes." CS230 Project Reports, Spring 2020, http://cs230.stanford.edu/projects_spring_2020/reports/38964602.pdf.

Yurko, Ryan, Samuel L. Ventura, and Maksim Horowitz. "Predicting Play Calls in the National Football League Using Hidden Markov Models." *IMA Journal of Management Mathematics*, vol. 32, no. 4, 2021, pp. 535-552. https://academic.oup.com/imaman/article/32/4/535/6211379?login=false.

Chan, Derek, et al. "Predicting Plays in the National Football League." *Journal of Sports Analytics*, vol. 6, no. 3, 2020, pp. 233-246. https://content.iospress.com/articles/journal-of-sports-analytics/jsa190348.

Jia, Z. and Li , Z. (2024) “Research on momentum quantification and influencing factors based on machine learning”, Transactions on Computer Science and Intelligent Systems Research, 5, pp. 1164–1171. doi:10.62051/z0rcmr69.https://wepub.org/index.php/TCSISR/article/view/2825

fairmodels: a Flexible Tool for Bias Detection, Visualization, and Mitigation. (2021). https://arxiv.org/abs/2104.00507

A Comprehensive Data Pipeline for Comparing the Effects of Momentum on Sports Outcomes. (2020). MDPI. https://www.mdpi.com/2306-5729/9/2/29

Appendix

Figure 1:

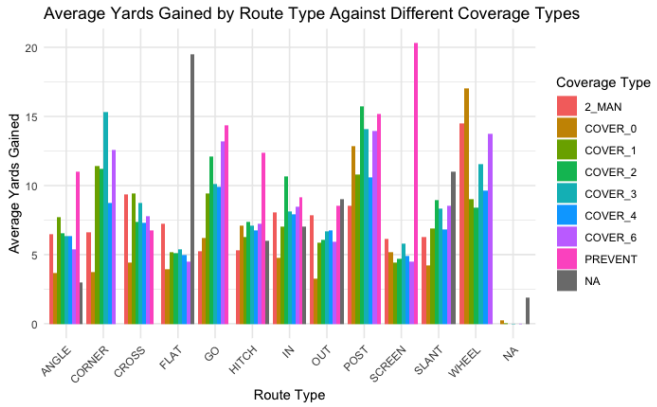


Figure 2:

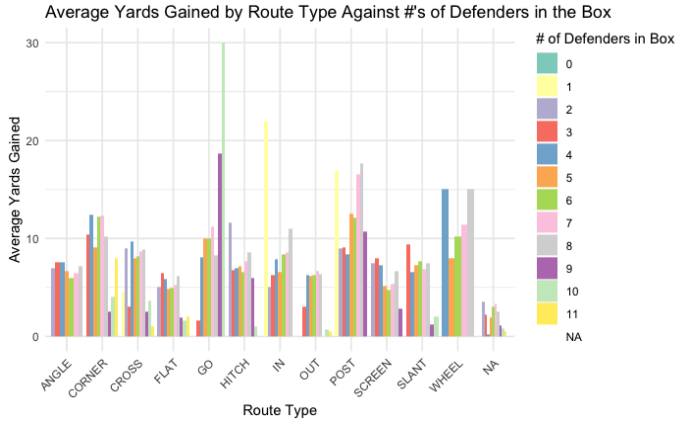


Figure 3:

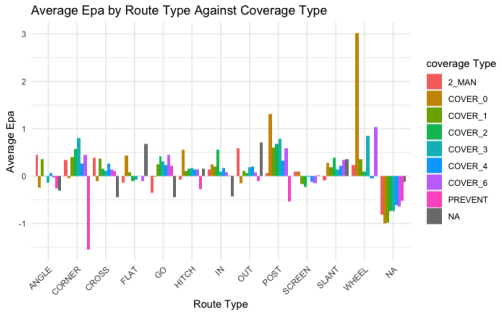


Figure 4:

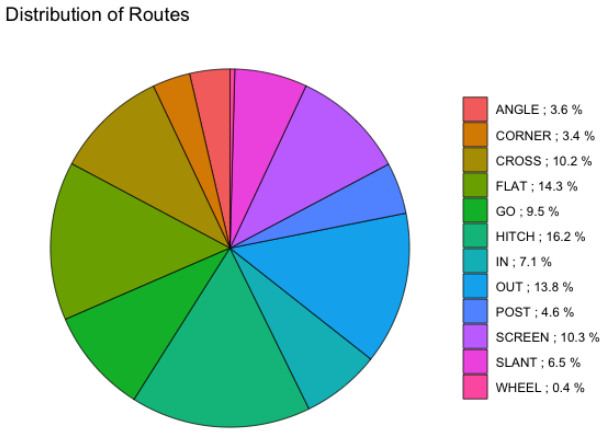


Figure 5:

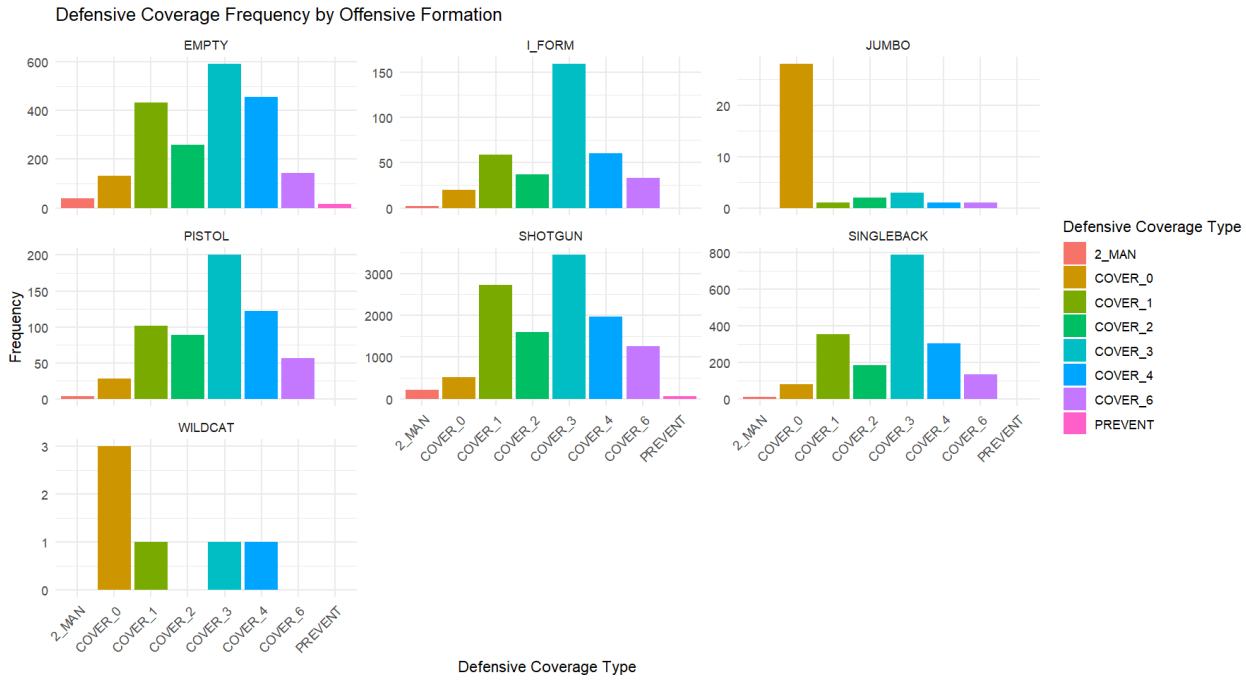


Figure 6:

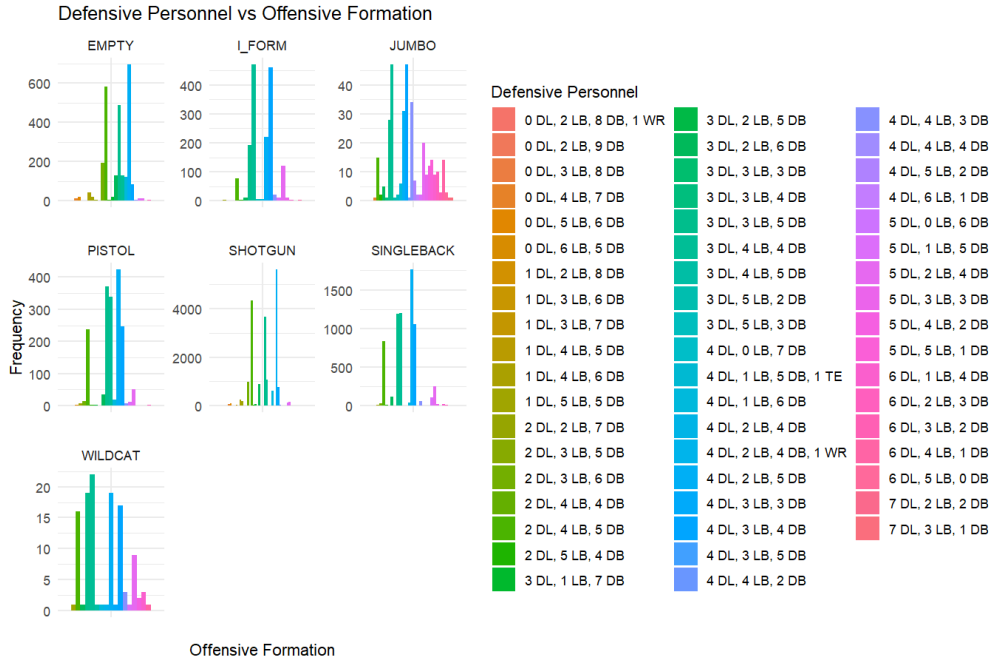


Figure 7:

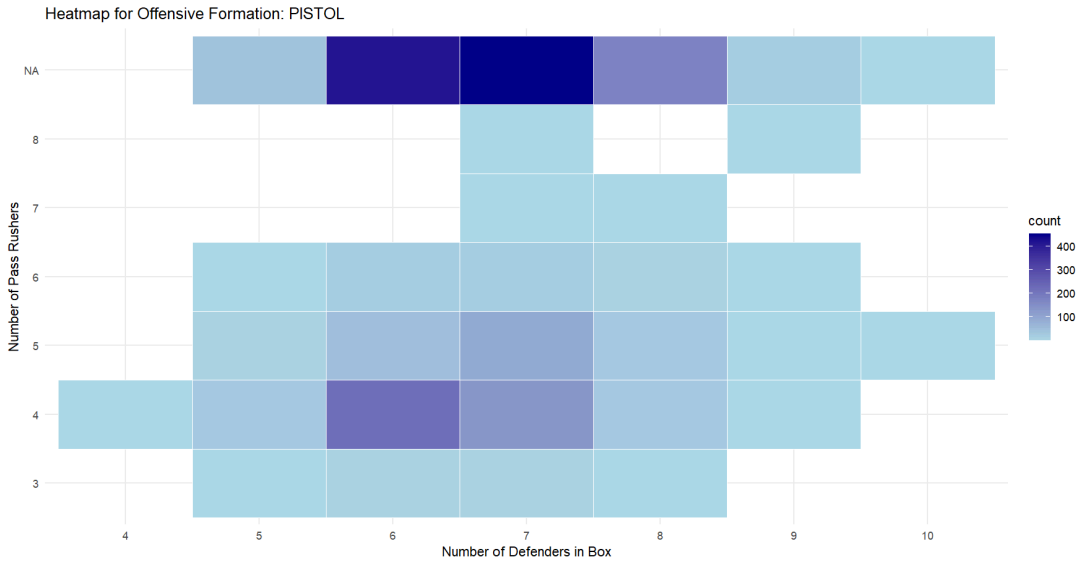


Figure 8:

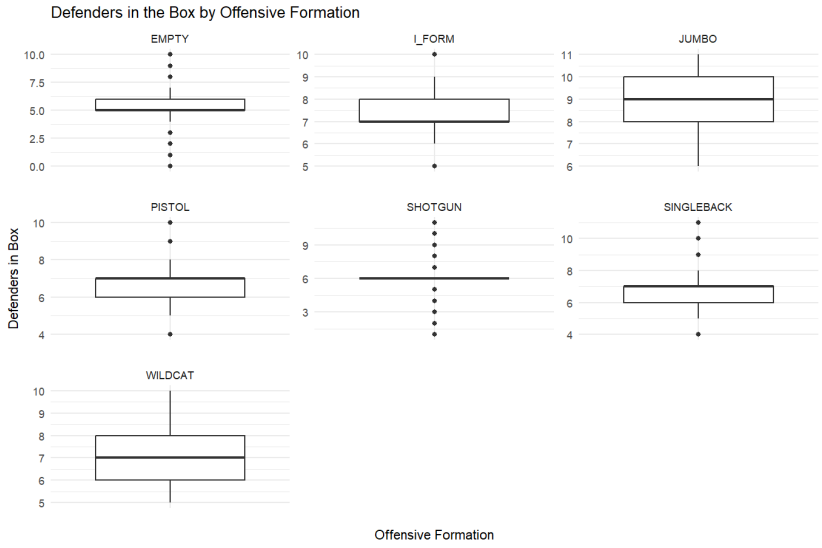


Figure 9:

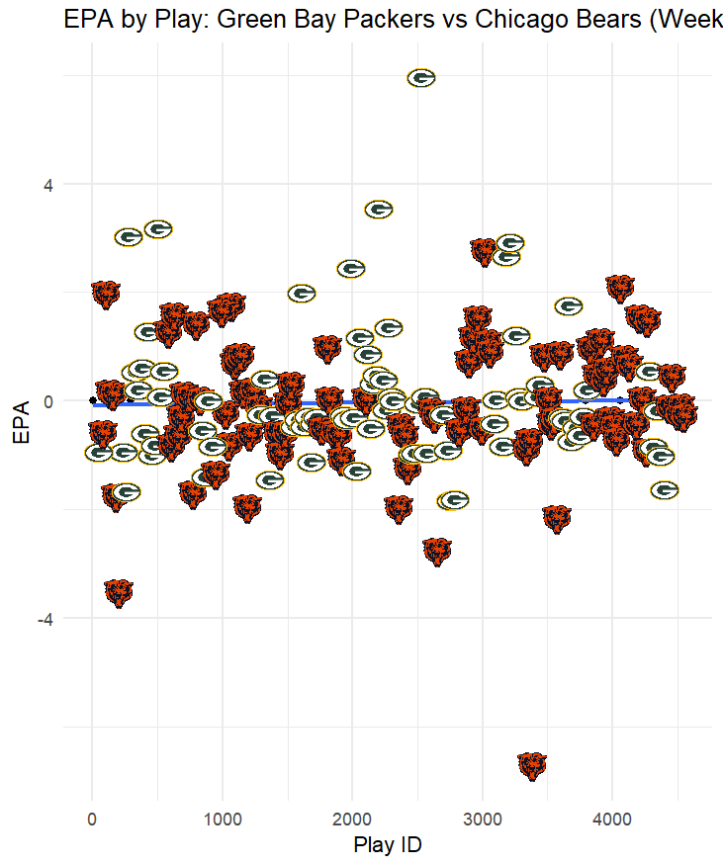


Figure 10:

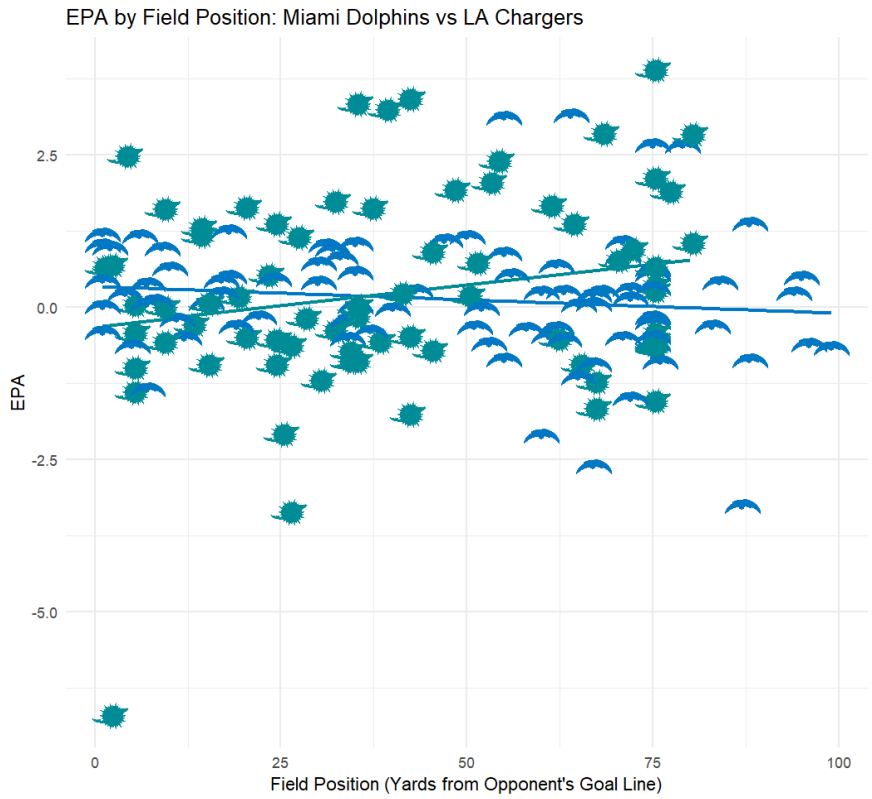


Figure 11:

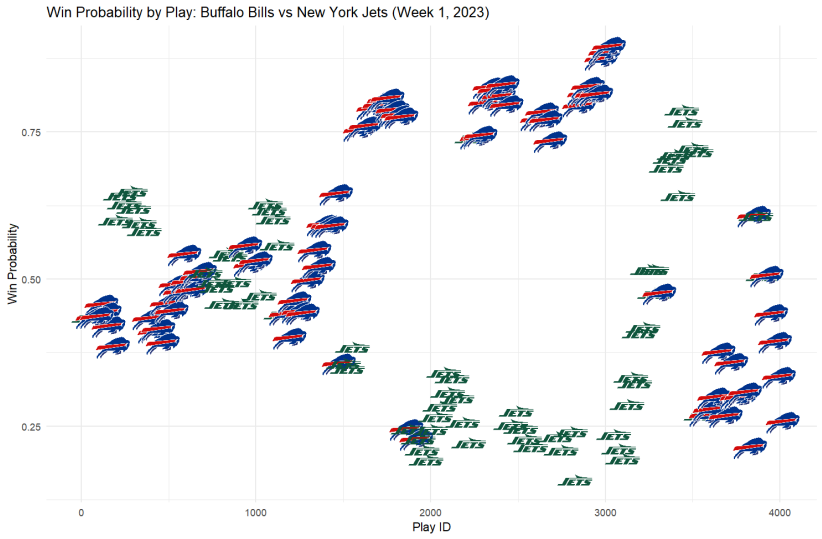


Figure 12:

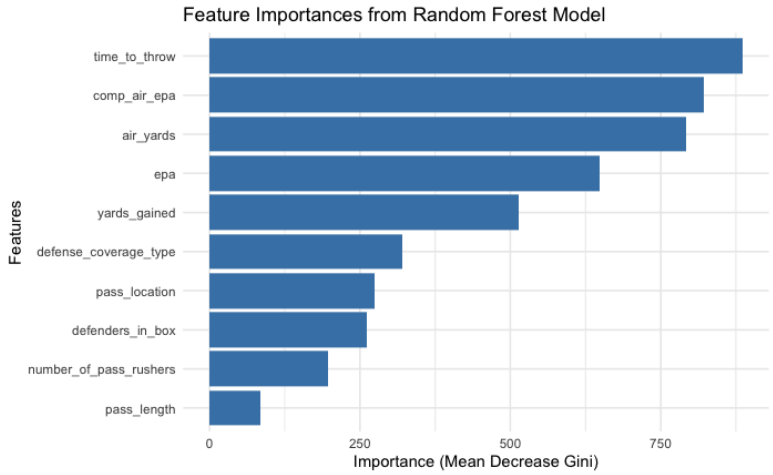


Figure 13:

Figure 14:

```
Overall Statistics

               Accuracy : 0.5003
                 95% CI : (0.4748, 0.5258)
    No Information Rate : 0.187
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4317
```

Figure 15:

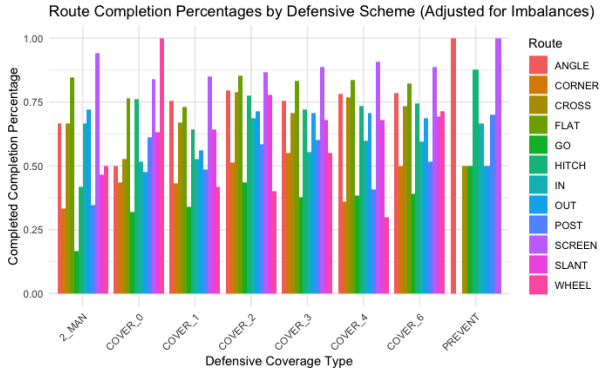


Figure 16:

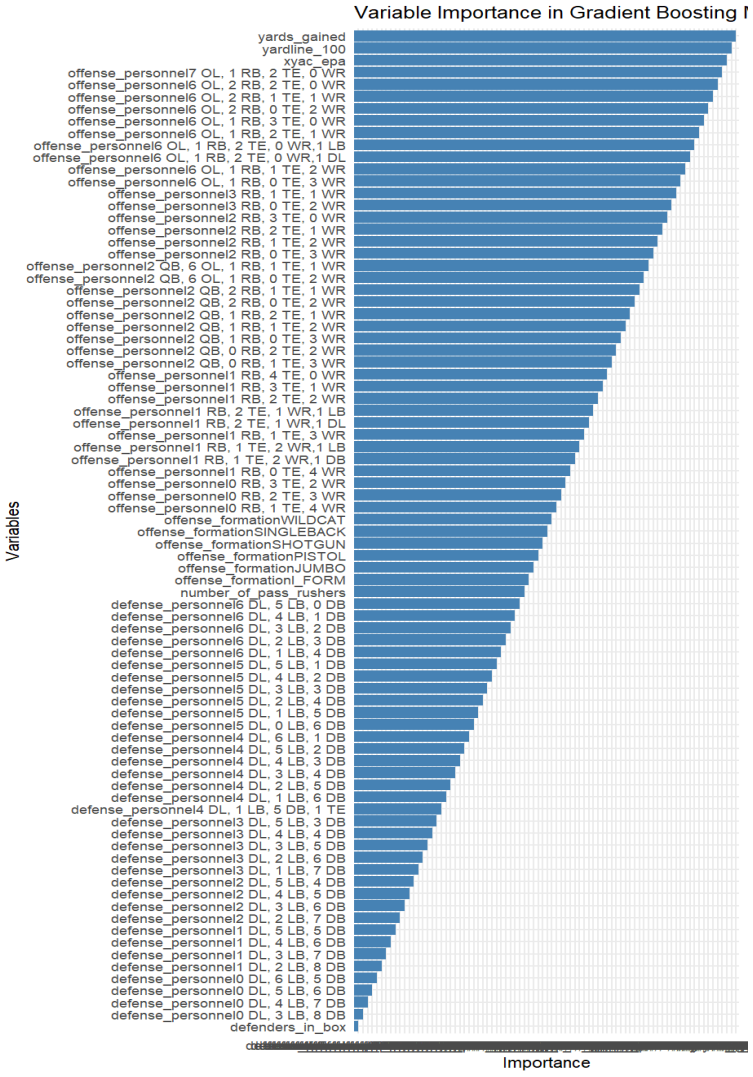


Figure 17:

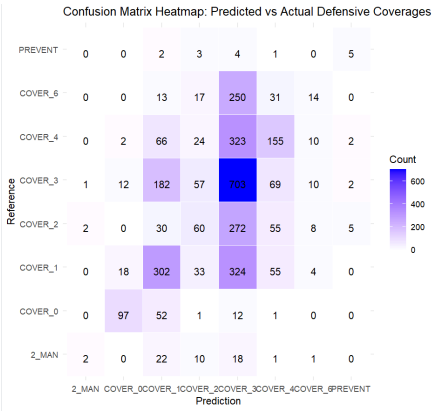


Figure 18:

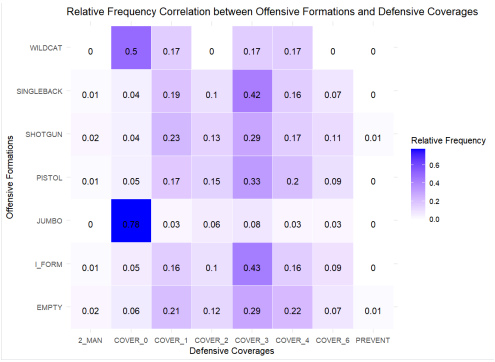


Figure 19:

```
> print(confusion_matrix)
Confusion Matrix and Statistics

          Reference
Prediction 2_MAN COVER_0 COVER_1 COVER_2 COVER_3 COVER_4
   2_MAN       1       0       1       2       2       0
   COVER_0     0      99      22       1      13       2
   COVER_1    22      50     293      34     174      67
   COVER_2    15       1      38      60      68      25
   COVER_3    14      12     318     265     682     307
   COVER_4     1       1      57      59      78     165
   COVER_6     1       0       7       8      17      14
   PREVENT     0       0       0       3       2       2
          Reference
Prediction COVER_6 PREVENT
   2_MAN         1       1
   COVER_0       0       0
   COVER_1      21       2
   COVER_2      23       4
   COVER_3     225       3
   COVER_4      37       1
   COVER_6      18       0
   PREVENT       0       4

Overall Statistics

               Accuracy : 0.3955
                 95% CI : (0.3788, 0.4123)
    No Information Rate : 0.3099
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1975

 Mcnemar's Test P-Value : NA

Statistics by Class:
```

Figure 20:

	Class: 2_MAN	Class: COVER_0
Sensitivity	0.0185185	0.60736
Specificity	0.9978717	0.98805
Pos Pred Value	0.1250000	0.72263
Neg Pred Value	0.9841079	0.98004
Prevalence	0.0161532	0.04876
Detection Rate	0.0002991	0.02961
Detection Prevalence	0.0023931	0.04098
Balanced Accuracy	0.5081951	0.79771
	Class: COVER_1	Class: COVER_2
Sensitivity	0.39810	0.13889
Specificity	0.85807	0.94023
Pos Pred Value	0.44193	0.25641
Neg Pred Value	0.83470	0.88035
Prevalence	0.22016	0.12923
Detection Rate	0.08765	0.01795
Detection Prevalence	0.19832	0.07000
Balanced Accuracy	0.62809	0.53956
	Class: COVER_3	Class: COVER_4
Sensitivity	0.6583	0.28351
Specificity	0.5041	0.91525
Pos Pred Value	0.3735	0.41353
Neg Pred Value	0.7666	0.85836
Prevalence	0.3099	0.17410
Detection Rate	0.2040	0.04936
Detection Prevalence	0.5462	0.11935
Balanced Accuracy	0.5812	0.59938
	Class: COVER_6	Class: PREVENT
Sensitivity	0.055385	0.266667
Specificity	0.984427	0.997897
Pos Pred Value	0.276923	0.363636
Neg Pred Value	0.906345	0.996699
Prevalence	0.097218	0.004487
Detection Rate	0.005384	0.001197
Detection Prevalence	0.019444	0.003290
Balanced Accuracy	0.519906	0.632282

Figure 21:

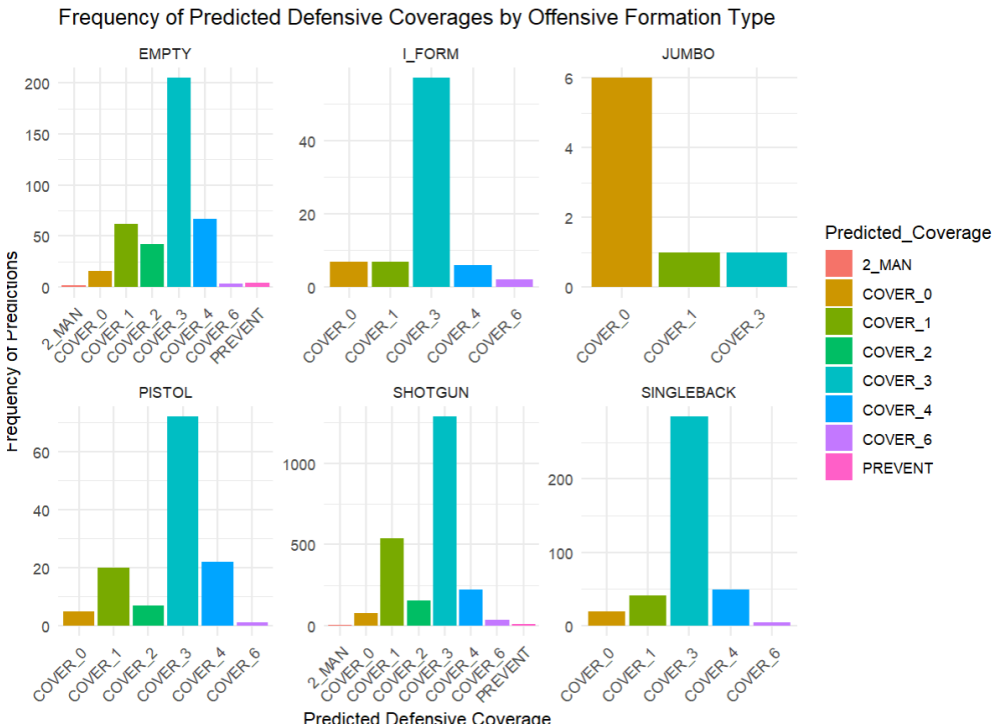


Figure 22:

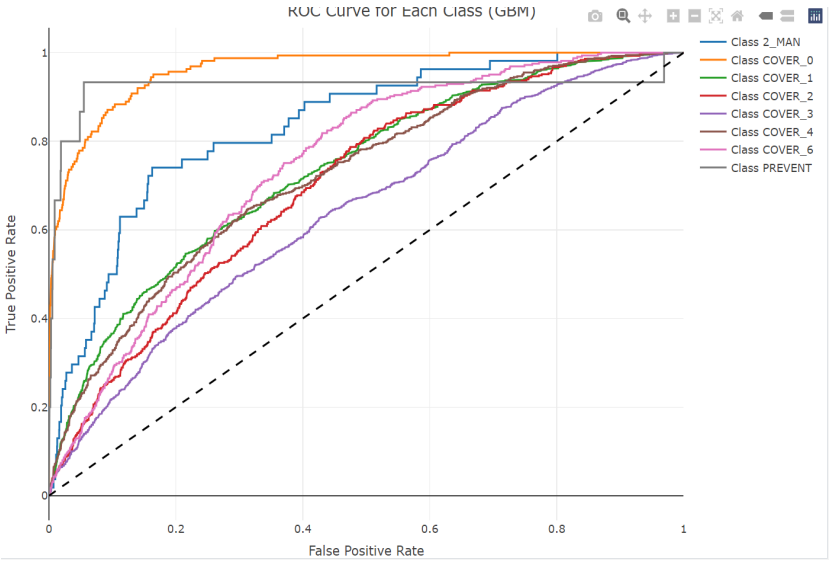


Figure 23:

```
Random Forest

31735 samples
   15 predictor

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 21157, 21157, 21156
Resampling results across tuning parameters:

  mtry  RMSE        Rsquared   MAE
  2     0.15050313  0.9630561  0.08145346
  4     0.06407545  0.9919411  0.02477951
  6     0.03760715  0.9970887  0.01146363

RMSE was used to select the optimal model using
 the smallest value.
The final value used for the model was mtry = 6.
```

Figure 24:

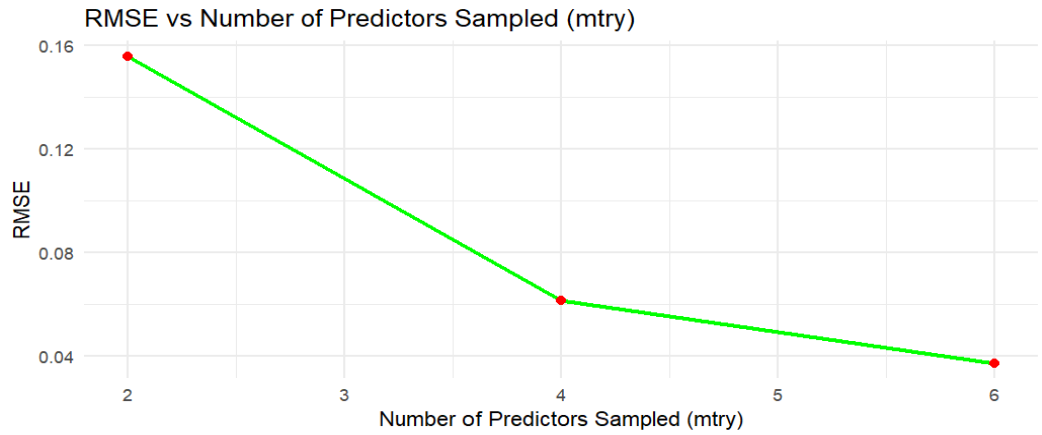


Figure 25:

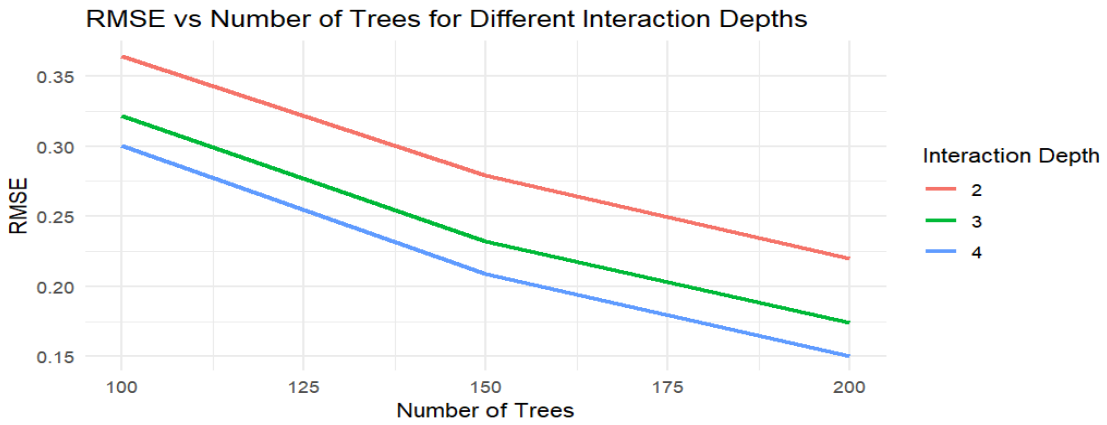


Figure 26:

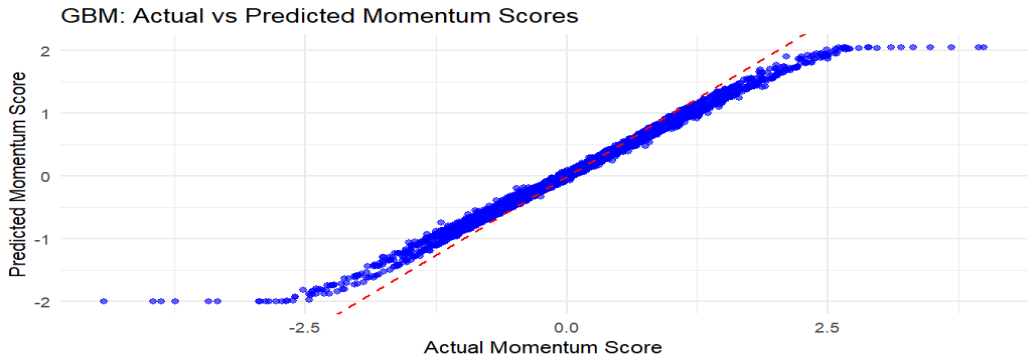
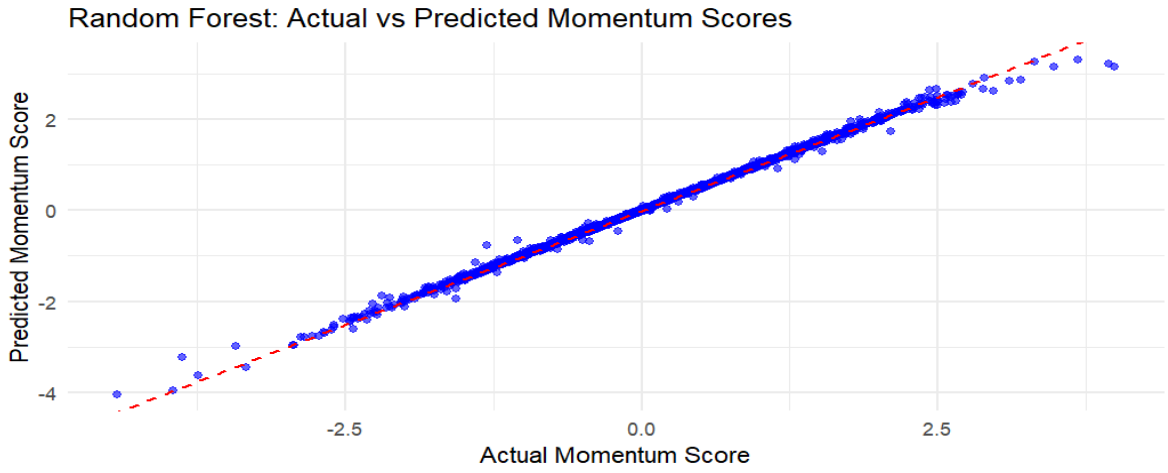


Figure 27:



Code:

RQ1:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(class)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
dim(gamez)
```

```
## [1] 43222   390
```

Preprocessing

```
data_rq1 <- gamez %>%
  filter(!is.na(route), play_type == "pass", (down == 1 & yards_gained/ydstogo >= .5) | (down == 2 & ya
  select(route, air_yards, yards_gained, defenders_in_box, number_of_pass_rushers, time_to_throw, defen
```

pass_length, pass_location,

```
dim(data_rq1)
```

```
## [1] 7588   11
```

```
sum(is.na(data_rq1))
```

```
## [1] 8
```

```
data_rq1 <- na.omit(data_rq1)
sum(is.na(data_rq1))
```

```
## [1] 0
```

```
dim(data_rq1)
```

```
## [1] 7580   11
```

```
head(data_rq1)
```

```
##   route air_yards yards_gained defenders_in_box number_of_pass_rushers
## 1 CROSS        15           19                6                      4
## 2 CROSS        16           17                6                      4
## 3 HITCH        12           12                6                      4
## 4 HITCH         5           12                6                      4
## 5 HITCH         1           10                8                      5
## 6 CROSS        13           13                7                      5
##   time_to_throw defense_coverage_type pass_length pass_location       epa
## 1         2.803               COVER_1       short        middle 1.6423130
## 2         3.803               COVER_4        deep         right 1.4982909
## 3         3.971               COVER_4       short        middle 1.1731545
## 4         2.102               COVER_4       short          left 0.7192315
## 5         2.870               COVER_3       short        middle 0.8752825
## 6         3.871               COVER_4       short          left 0.7226735
##   comp_air_epa
## 1    1.4033604
## 2    1.4388244
## 3    1.1731545
## 4    0.0726840
## 5   -0.4602623
## 6    0.7226735
```

Encoding and Scaling

```
data_rq1$pass_length <- as.numeric(factor(data_rq1$pass_length, levels = c('short','deep')))
```

```
data_rq1$pass_location <- as.numeric(factor(data_rq1$pass_location))
```

```
data_rq1$defense_coverage_type <- as.numeric(factor(data_rq1$defense_coverage_type))
```

```
unique(data_rq1$defense_coverage_type)
```

```
## [1] 3 6 5 7 4 2 1 8
```

```
data_rq1$yards_gained <- scale(data_rq1$yards_gained)
data_rq1$defenders_in_box <- scale(data_rq1$defenders_in_box)
data_rq1$epa <- scale(data_rq1$epa)
data_rq1$air_yards <- scale(data_rq1$air_yards)
data_rq1$number_of_pass_rushers <- scale(data_rq1$number_of_pass_rushers)
data_rq1$time_to_throw <- scale(data_rq1$time_to_throw)
data_rq1$comp_air_epa <- scale(data_rq1$comp_air_epa)
```

Create Models

```
set.seed(42)
train_index <- createDataPartition(data_rq1$route, p = 0.8, list = FALSE)
train_data <- data_rq1[train_index, ]
test_data <- data_rq1[-train_index, ]
```

```
rf_model <- randomForest(as.factor(route) ~ ., data = train_data,
                         ntree = 500,
                         mtry = 2,
                         importance = TRUE)
```

Accuracy Metrics and Confusion Matrix

```
predictions <- predict(rf_model, test_data)

confusionMatrix(predictions, as.factor(test_data$route))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ANGLE CORNER CROSS FLAT  GO HITCH  IN OUT POST SCREEN SLANT WHEEL
##      ANGLE     17      0    10    1   0     7   1   3    0      0     1     0
##      CORNER     0     11     8    0   8     1   0   5    2      0     0     1
##      CROSS     11      7    70   16   2    17  10  16    1      1     5     1
##      FLAT       9      2    23  127   1     3   1  18    1     28     2     0
##      GO         0     13     0    0  70     4   1   8   12      0     2     1
##      HITCH     10      7    23   14   3   169  30  95    2      0    35     2
```

```
##      IN         2      2     5    0   0     7  42   2   21      0     5     0
##      OUT        3      7    15    5  10    58  18  67    4      0     8     1
##      POST       0      1     3    0  11     1   5   1   29      0     1     0
##      SCREEN     2      0     6   25   1     0   0   1    0     95     0     0
##      SLANT      0      0     5    0   5    16   5   9    6      0    54     0
##      WHEEL      0      0     0    0   0     0   0   0    0      0     0     0
##
## Overall Statistics
##
##                Accuracy : 0.4964
##                  95% CI : (0.4709, 0.5219)
##     No Information Rate : 0.187
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4265
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: ANGLE Class: CORNER Class: CROSS Class: FLAT
## Sensitivity               0.31481       0.22000      0.41667     0.67553
## Specificity               0.98424       0.98291      0.93532     0.93358
## Pos Pred Value            0.42500       0.30556      0.44586     0.59070
## Neg Pred Value            0.97488       0.97360      0.92773     0.95300
## Prevalence                0.03569       0.03305      0.11104     0.12426
## Detection Rate            0.01124       0.00727      0.04627     0.08394
## Detection Prevalence      0.02644       0.02379      0.10377     0.14210
## Balanced Accuracy         0.64953       0.60146      0.67599     0.80456
##                      Class: GO Class: HITCH Class: IN Class: OUT Class: POST
## Sensitivity            0.63063       0.5972   0.37168     0.29778     0.37179
## Specificity            0.97076       0.8203   0.96857     0.89984     0.98397
## Pos Pred Value         0.63063       0.4333   0.48837     0.34184     0.55769
## Neg Pred Value         0.97076       0.8985   0.95025     0.88003     0.96646
## Prevalence             0.07336       0.1870   0.07469     0.14871     0.05155
## Detection Rate         0.04627       0.1117   0.02776     0.04428     0.01917
## Detection Prevalence   0.07336       0.2578   0.05684     0.12954     0.03437
## Balanced Accuracy      0.80069       0.7087   0.67013     0.59881     0.67788
##                      Class: SCREEN Class: SLANT Class: WHEEL
## Sensitivity                0.76613      0.47788     0.000000
## Specificity                0.97480      0.96714     1.000000
## Pos Pred Value             0.73077      0.54000          NaN
## Neg Pred Value             0.97903      0.95824     0.996034
## Prevalence                 0.08196      0.07469     0.003966
## Detection Rate             0.06279      0.03569     0.000000
## Detection Prevalence       0.08592      0.06609     0.000000
## Balanced Accuracy          0.87047      0.72251     0.500000
```

```r
train_control <- trainControl(method = "cv", number = 5, search = "grid")


tune_grid <- expand.grid(mtry = c(1, 2, 3, 4, 5))

rf_tuned <- train(as.factor(route) ~ .,
```

```
                  data = train_data,
                  method = "rf",
                  trControl = train_control,
                  tuneGrid = tune_grid,
                  ntree = 500,
                  importance = TRUE)

predictions_tuned <- predict(rf_tuned, test_data)

confusionMatrix(predictions_tuned, as.factor(test_data$route))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ANGLE CORNER CROSS FLAT  GO HITCH  IN OUT POST SCREEN SLANT WHEEL
##     ANGLE     20      0    10    1   0     7   1   3    0      0     1     0
##     CORNER     0     13     8    0   9     1   2   6    4      0     0     1
##     CROSS      8      7    66   14   3    16  10  17    0      1     5     1
##     FLAT       9      2    24  125   2     5   1  18    1     26     3     0
##     GO         0     11     1    0  68     3   0   7   13      0     2     1
##     HITCH     10      5    23   16   2   168  29  92    3      0    38     2
##     IN         2      4     5    0   0    10  42   3   20      0     6     0
##     OUT        3      7    15    6  10    54  17  68    2      0     6     1
##     POST       0      1     4    0  11     1   5   1   27      0     1     0
##     SCREEN     2      0     6   26   1     0   0   1    0     97     0     0
##     SLANT      0      0     6    0   5    18   6   9    8      0    51     0
##     WHEEL      0      0     0    0   0     0   0   0    0      0     0     0
##
## Overall Statistics
##
##                Accuracy : 0.4924
##                  95% CI : (0.4669, 0.5179)
##     No Information Rate : 0.187
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4226
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: ANGLE Class: CORNER Class: CROSS Class: FLAT
## Sensitivity               0.37037      0.260000      0.39286     0.66489
## Specificity               0.98424      0.978811      0.93903     0.93132
## Pos Pred Value            0.46512      0.295455      0.44595     0.57870
## Neg Pred Value            0.97687      0.974813      0.92527     0.95143
## Prevalence                0.03569      0.033047      0.11104     0.12426
## Detection Rate            0.01322      0.008592      0.04362     0.08262
## Detection Prevalence      0.02842      0.029081      0.09782     0.14276
## Balanced Accuracy         0.67730      0.619405      0.66595     0.79811
##                      Class: GO Class: HITCH Class: IN Class: OUT Class: POST
```

```
## Sensitivity            0.61261       0.5936   0.37168    0.30222     0.34615
## Specificity            0.97290       0.8211   0.96429    0.90606     0.98328
## Pos Pred Value         0.64151       0.4330   0.45652    0.35979     0.52941
## Neg Pred Value         0.96944       0.8978   0.95004    0.88142     0.96512
## Prevalence             0.07336       0.1870   0.07469    0.14871     0.05155
## Detection Rate         0.04494       0.1110   0.02776    0.04494     0.01785
## Detection Prevalence   0.07006       0.2564   0.06081    0.12492     0.03371
## Balanced Accuracy      0.79275       0.7074   0.66798    0.60414     0.66471
##                      Class: SCREEN Class: SLANT Class: WHEEL
## Sensitivity                0.78226      0.45133      0.000000
## Specificity                0.97408      0.96286      1.000000
## Pos Pred Value             0.72932      0.49515          NaN
## Neg Pred Value             0.98043      0.95603      0.996034
## Prevalence                 0.08196      0.07469      0.003966
## Detection Rate             0.06411      0.03371      0.000000
## Detection Prevalence       0.08790      0.06808      0.000000
## Balanced Accuracy          0.87817      0.70709      0.500000
```

```
conf_matrix <- confusionMatrix(predictions_tuned, as.factor(test_data$route))
conf_table <- conf_matrix$table
print(conf_table)
```

```
##           Reference
## Prediction ANGLE CORNER CROSS FLAT  GO HITCH  IN OUT POST SCREEN SLANT WHEEL
##      ANGLE     20      0    10    1   0     7   1   3    0      0     1     0
##      CORNER     0     13     8    0   9     1   2   6    4      0     0     1
##      CROSS      8      7    66   14   3    16  10  17    0      1     5     1
##      FLAT       9      2    24  125   2     5   1  18    1     26     3     0
##      GO         0     11     1    0  68     3   0   7   13      0     2     1
##      HITCH     10      5    23   16   2   168  29  92    3      0    38     2
##      IN         2      4     5    0   0    10  42   3   20      0     6     0
##      OUT        3      7    15    6  10    54  17  68    2      0     6     1
##      POST       0      1     4    0  11     1   5   1   27      0     1     0
##      SCREEN     2      0     6   26   1     0   0   1    0     97     0     0
##      SLANT      0      0     6    0   5    18   6   9    8      0    51     0
##      WHEEL      0      0     0    0   0     0   0   0    0      0     0     0
```

```
dim(conf_table)
```

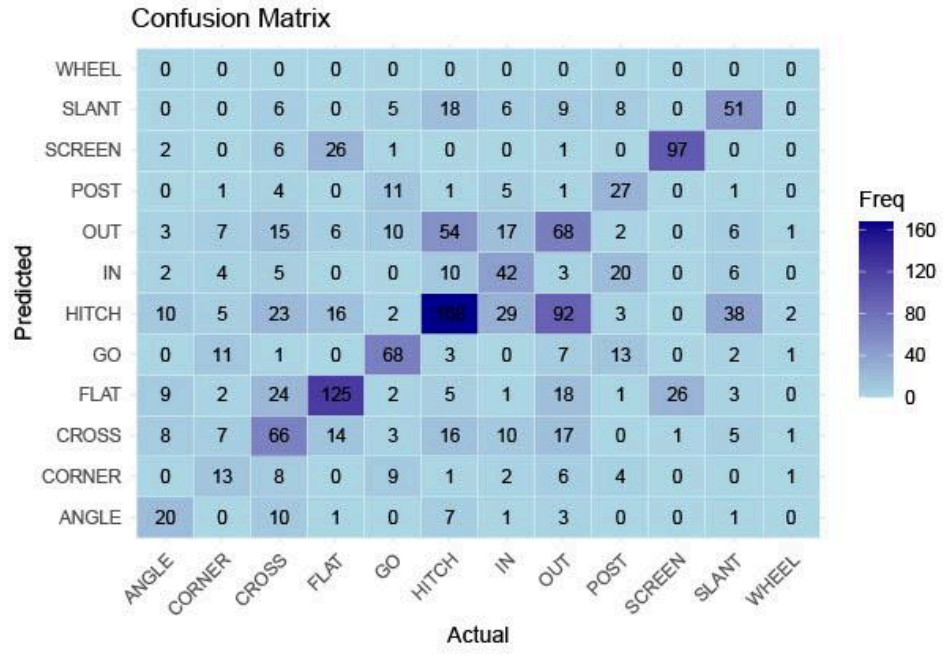
```
## [1] 12 12
```

Visualizations

```
cm_df <- as.data.frame(conf_table)

# Plot using ggplot2
ggplot(cm_df, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Color gradient
  geom_text(aes(label = Freq), color = "black", size = 3) +  # Frequency count labels
```

```
  labs(title = "Confusion Matrix", x = "Actual", y = "Predicted") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
new_subset <- gamez %>%
filter(!is.na(route), !is.na(defense_coverage_type), play_type == "pass", (down == 1 & yards_gained/yds
  select(route, air_yards, yards_gained, defenders_in_box, number_of_pass_rushers, time_to_throw, defen
```

```
most_effective_routes <- new_subset %>%
  group_by(defense_coverage_type, route) %>%
  summarize(completed_route_percentage = sum(complete_pass ==1)/n(), .groups = "drop") %>%
  arrange(defense_coverage_type, desc(completed_route_percentage))
```

```
ggplot(most_effective_routes, aes(x = defense_coverage_type, y = completed_route_percentage, fill = rou
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Route Completion Percentages by Defensive Scheme (Adjusted for Imbalances)",
       x = "Defensive Coverage Type",
       y = "Completed Completion Percentage",
       fill = "Route") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

RQ2:

```r
# Load necessary libraries
library(tidyverse)
library(caret)
library(gbm)
library(pROC)
library(plotly)
library(pdp)
library(reshape2)
library(ggplot2)

# Load the dataset
data <- read.csv("gamez_data.csv")

# Select relevant variables and clean the data
model_data <- data %>%
  select(defense_coverage_type, offense_formation, yards_gained, yardline_100,
         offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers, xyac_epa) %>%
  drop_na(defense_coverage_type, offense_formation, offense_personnel, defense_personnel)

# Convert defense_coverage_type to factor
model_data$defense_coverage_type <- as.factor(model_data$defense_coverage_type)

# Display the levels of defense_coverage_type along with their numeric codes
defense_coverage_levels <- levels(model_data$defense_coverage_type)
coverage_codes <- seq_along(defense_coverage_levels)
coverage_mapping <- data.frame(Code = coverage_codes, Coverage = defense_coverage_levels)
print("Mapping of Codes to Coverage Types:")
print(coverage_mapping)

# Identify the coverage types corresponding to the problematic class numbers
problematic_codes <- c(19, 20, 22, 27, 28)
problematic_coverages <- defense_coverage_levels[problematic_codes]
cat("Problematic Coverage Types to be Removed:\n")
print(problematic_coverages)

# Filter out observations with the problematic coverage types
model_data <- model_data[!(model_data$defense_coverage_type %in% problematic_coverages), ]

# Drop unused factor levels after filtering
model_data$defense_coverage_type <- droplevels(model_data$defense_coverage_type)

# Proceed with the rest of your data processing and modeling
# Fill missing numeric values with the median
model_data$defenders_in_box <- ifelse(is.na(model_data$defenders_in_box),
                                      median(model_data$defenders_in_box, na.rm = TRUE),
                                      model_data$defenders_in_box)
model_data$number_of_pass_rushers <- ifelse(is.na(model_data$number_of_pass_rushers),
                                            median(model_data$number_of_pass_rushers, na.rm = TRUE),
                                            model_data$number_of_pass_rushers)
model_data$xyac_epa <- ifelse(is.na(model_data$xyac_epa),
                              median(model_data$xyac_epa, na.rm = TRUE),
                              model_data$xyac_epa)

# Convert other categorical variables to factors
model_data$offense_formation <- as.factor(model_data$offense_formation)
model_data$offense_personnel <- as.factor(model_data$offense_personnel)
model_data$defense_personnel <- as.factor(model_data$defense_personnel)

# Split the data into training and test sets
set.seed(123)
train_index <- createDataPartition(model_data$defense_coverage_type, p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]

# Set up cross-validation for gradient boosting tuning
train_control <- trainControl(method = "cv", number = 3)

# Define a simpler grid of hyperparameters to speed up training
gbm_grid <- expand.grid(
  interaction.depth = c(3),
  n.trees = c(100),
  shrinkage = c(0.01, 0.1),
  n.minobsinnode = 10
)

# Train the GBM model
set.seed(123)
tuned_gbm <- train(
  defense_coverage_type ~ offense_formation + yards_gained + yardline_100 +
    offense_personnel + defenders_in_box +
    defense_personnel + number_of_pass_rushers + xyac_epa,
  data = train_data,
  method = "gbm",
  trControl = train_control,
  tuneGrid = gbm_grid,
  verbose = FALSE
)

# Make predictions on the test set
gbm_predictions <- predict(tuned_gbm, newdata = test_data)

# Evaluate model performance using confusion matrix and accuracy
confusion_matrix <- confusionMatrix(gbm_predictions, test_data$defense_coverage_type)
print(confusion_matrix)

# Output the accuracy
accuracy <- confusion_matrix$overall['Accuracy']
cat("Gradient Boosting Model Accuracy:", accuracy, "\n")

# Add the predicted coverages to the test dataset
test_data$Predicted_Coverage <- gbm_predictions
```

```r
test_data$Predicted_Coverage <- gbm_predictions

# Count the frequency of each predicted coverage by offensive formation type
coverage_counts <- test_data %>%
  group_by(offense_formation, Predicted_Coverage) %>%
  summarise(Frequency = n()) %>%
  ungroup()

# Plot separate bar plots for each offensive formation type using facet_wrap
ggplot(coverage_counts, aes(x = Predicted_Coverage, y = Frequency, fill = Predicted_Coverage)) +
  geom_bar(stat = "identity") +
  labs(title = "Frequency of Predicted Defensive Coverages by Offensive Formation Type",
       x = "Predicted Defensive Coverage",
       y = "Frequency of Predictions") +
  facet_wrap(~ offense_formation, scales = "free") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

RQ3:

```r
# Load necessary libraries
library(tidyverse)
library(caret)
library(randomForest)
library(gbm)
library(car)  # For VIF calculation

# Load the dataset
data <- read.csv("gamez_data.csv")

# Filter and clean the data for RQ3 subset
momentum_data <- data %>%
  select(score_differential, td_prob, fg_prob, epa, wpa, yards_gained,
         drive_ended_with_score, interception, fumble_lost, total_home_score, total_away_score, yardline_100) %>%
  drop_na()

# Create new variables for momentum score calculation
momentum_data <- momentum_data %>%
  mutate(
    delta_wpa = c(NA, diff(wpa)),
    delta_epa = c(NA, diff(epa)),
    scoring_event = ifelse(drive_ended_with_score == 1, 1, 0),
    turnover_event = ifelse(interception == 1 | fumble_lost == 1, -1, 0)
  ) %>%
  drop_na()  # Drop NAs introduced by diff

# Define the Momentum Score (MS) with finely tuned weights
momentum_data <- momentum_data %>%
  mutate(
    momentum_score = 0.45 * delta_wpa + 0.35 * delta_epa + 0.15 * scoring_event + 0.05 * turnover_event
  )

# Identify multicollinearity using correlation matrix
cor_matrix <- cor(momentum_data %>% select(-momentum_score))  # Exclude the target variable
high_corr <- findCorrelation(cor_matrix, cutoff = 0.9)  # Find highly correlated variables with correlation > 0.9
```

```r
high_corr <- findCorrelation(cor_matrix, cutoff = 0.9)  # Find highly correlated variables with correlation > 0.9
momentum_data <- momentum_data %>% select(-all_of(high_corr))  # Remove highly correlated variables

# Split the dataset into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(momentum_data$momentum_score, p = 0.8, list = FALSE)
train_data <- momentum_data[trainIndex, ]
test_data <- momentum_data[-trainIndex, ]

# ---------- Random Forest Model Tuning and Evaluation ---------- #

# Random Forest model tuning
rf_grid <- expand.grid(mtry = c(2, 4, 6))  # Smaller grid for tuning
rf_model_tuned <- train(
  momentum_score ~ .,
  data = train_data,
  method = "rf",
  trControl = trainControl(method = "cv", number = 3, savePredictions = TRUE),  # Reduced to 3-fold CV
  tuneGrid = rf_grid,  # Smaller tuning grid
  ntree = 200  # Reduced number of trees
)

# Predict and evaluate Random Forest model
rf_predictions <- predict(rf_model_tuned, newdata = test_data)
rf_rmse <- RMSE(rf_predictions, test_data$momentum_score)
rf_mae <- MAE(rf_predictions, test_data$momentum_score)
cat("Random Forest RMSE:", rf_rmse, "\n")
cat("Random Forest MAE:", rf_mae, "\n")

# ---------- GBM Model Tuning and Evaluation ---------- #

# GBM model tuning
gbm_grid <- expand.grid(
  interaction.depth = c(2, 3, 4),    # Expanded grid for interaction depth
  n.trees = c(100, 150, 200),        # Vary number of trees for better tuning
  shrinkage = 0.01,                  # Fixed shrinkage rate
  n.minobsinnode = 10                # Fixed minimum observations in node
)

gbm_model_tuned <- train(
  momentum_score ~ .,
  data = train_data,
  method = "gbm",
  trControl = trainControl(method = "cv", number = 3),  # Reduced to 3-fold CV
  verbose = FALSE,
  tuneGrid = gbm_grid  # Expanded grid with multiple values for tuning
)

# Predict and evaluate GBM model
gbm_predictions <- predict(gbm_model_tuned, newdata = test_data)
gbm_rmse <- RMSE(gbm_predictions, test_data$momentum_score)
gbm_mae <- MAE(gbm_predictions, test_data$momentum_score)
cat("GBM RMSE:", gbm_rmse, "\n")
cat("GBM MAE:", gbm_mae, "\n")

# ---------- Visualization for Random Forest ---------- #

# 1. R-Squared vs Number of Predictors Sampled (mtry)
rf_results <- rf_model_tuned$results
ggplot(rf_results, aes(x = mtry, y = Rsquared)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "R-Squared vs Number of Predictors Sampled (mtry)",
       x = "Number of Predictors Sampled (mtry)",
       y = "R-Squared") +
  theme_minimal()

# 2. RMSE vs Number of Predictors Sampled (mtry)
ggplot(rf_results, aes(x = mtry, y = RMSE)) +
  geom_line(color = "green", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "RMSE vs Number of Predictors Sampled (mtry)",
       x = "Number of Predictors Sampled (mtry)",
       y = "RMSE") +
  theme_minimal()

# 3. Feature Importance (Random Forest)
importance_df <- varImp(rf_model_tuned$finalModel)
importance_df <- data.frame(Feature = rownames(importance_df), Importance = importance_df$Overall)
ggplot(importance_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Feature Importance (Random Forest)",
       x = "Feature", y = "Importance") +
  theme_minimal()

# 4. Actual vs Predicted (Random Forest)
comparison_rf_df <- data.frame(Actual = test_data$momentum_score, Predicted = rf_predictions)
ggplot(comparison_rf_df, aes(x = Actual, y = Predicted)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  # Reference line
  labs(title = "Random Forest: Actual vs Predicted Momentum Scores",
       x = "Actual Momentum Score", y = "Predicted Momentum Score") +
  theme_minimal()

# 5. Residuals vs Predicted (Random Forest)
residuals_rf <- comparison_rf_df$Actual - comparison_rf_df$Predicted
ggplot(data.frame(Residuals = residuals_rf, Predicted = comparison_rf_df$Predicted), aes(x = Predicted, y = Residuals)) +
  geom_point(color = "purple", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Predicted (Random Forest)",
       x = "Predicted Momentum Score", y = "Residuals") +
  theme_minimal()
```

```r

# ---------- Visualization for GBM ---------- #

# 1. R-Squared vs Number of Trees for Different Depths
gbm_results <- gbm_model_tuned$results
ggplot(gbm_results, aes(x = n.trees, y = Rsquared, color = as.factor(interaction.depth))) +
  geom_line(size = 1) +
  labs(title = "R-Squared vs Number of Trees for Different Interaction Depths",
       x = "Number of Trees",
       y = "R-Squared",
       color = "Interaction Depth") +
  theme_minimal()

# 2. RMSE vs Number of Trees for Different Depths
ggplot(gbm_results, aes(x = n.trees, y = RMSE, color = as.factor(interaction.depth))) +
  geom_line(size = 1) +
  labs(title = "RMSE vs Number of Trees for Different Interaction Depths",
       x = "Number of Trees",
       y = "RMSE",
       color = "Interaction Depth") +
  theme_minimal()

# 3. Feature Importance (GBM)
importance_gbm <- summary(gbm_model_tuned$finalModel, plot = FALSE)
importance_gbm_df <- data.frame(Feature = rownames(importance_gbm), Importance = importance_gbm[,1])
ggplot(importance_gbm_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(title = "Feature Importance (GBM)",
       x = "Features", y = "Relative Importance") +
  theme_minimal()

# 4. Actual vs Predicted (GBM)
# 4. Actual vs Predicted (GBM)
comparison_gbm_df <- data.frame(Actual = test_data$momentum_score, Predicted = gbm_predictions)
```

```r
ggplot(comparison_gbm_df, aes(x = Actual, y = Predicted)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  # Reference line
  labs(title = "GBM: Actual vs Predicted Momentum Scores",
       x = "Actual Momentum Score", y = "Predicted Momentum Score") +
  theme_minimal()

# 5. Residuals vs Predicted (GBM)
residuals_gbm <- comparison_gbm_df$Actual - comparison_gbm_df$Predicted
ggplot(data.frame(Residuals = residuals_gbm, Predicted = comparison_gbm_df$Predicted), aes(x = Predicted, y = Residuals)) +
  geom_point(color = "purple", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Predicted (GBM)",
       x = "Predicted Momentum Score", y = "Residuals") +
  theme_minimal()

# Comparison of Model Performances
model_performance <- data.frame(
  Model = c("Random Forest", "GBM"),
  RMSE = c(rf_rmse, gbm_rmse),
  MAE = c(rf_mae, gbm_mae)
)
print(model_performance)


# End of code
```