Predicting NFL Game Outcomes Using Ensemble Machine Learning and Efficiency-Based Metrics

Johnathan Gutierrez-Diaz
CAI-5107
University of South Florida
Jmgutierrezdiaz@usf.edu

Abstract—**Predicting outcomes in the National Football League (NFL) is a challenging task due to the sport's high variability, strategic complexity, and frequent roster changes. This study presents a data-driven framework for forecasting NFL game results using machine learning models trained on team-level offensive and defensive efficiency metrics. A dataset of 7,158 regular-season games from 1999 through 2025 was constructed using publicly available play-by-play data with advanced performance statistics, including expected points added (EPA), quarterback efficiency, pressure rates, and explosiveness metrics for both home and away teams. Three models were evaluated: logistic regression, random forest, and XGBoost. Hyperparameter optimization was performed using Bayesian search, and performance was measured on a held-out 30% test set. Results indicate that tree-based ensemble methods outperform logistic regression, with tuned XGBoost achieving the best performance (accuracy = 0.876, AUROC = 0.952). To assess temporal robustness, models were additionally validated on post-2019 data, where a voting ensemble achieved a 91.99% simulated wagering accuracy. The framework demonstrates that efficiency-based metrics combined with ensemble learning provide robust and generalizable NFL game predictions, with practical implications for sports analytics and forecasting applications.**

I. Introduction

The National Football League (NFL) represents one of the most competitive and unpredictable professional sports leagues, where game outcomes carry significant implications for betting markets, team decision-making, and fan engagement. Accurate game-level prediction remains a challenging problem due to the dynamic nature of team performance, injuries, coaching strategies, and situational factors that influence match outcomes each week. As professional football continues to evolve toward a data-driven industry, advanced analytics and machine learning models have become essential tools for capturing the complex relationships embedded within team-level and player-level statistics.

In this project, the original predictive framework developed in Milestone I has been substantially expanded through both feature engineering and dataset refinement. In addition to traditional offensive statistics, this study incorporates advanced efficiency metrics, including expected points added (EPA) derived from passing and rushing plays, along with situational metrics such as home and away EPA splits. Defensive performance is also modeled in greater detail, with features including pressure rates, yards allowed, turnovers forced, and opponent efficiency statistics. These additions aim to better capture the hidden influence of defensive dominance and game-flow efficiency, two aspects that are often underrepresented in conventional statistical approaches.

Furthermore, the final version of this project restricts training data to the 1999–present era of the NFL, an intentional design choice reflecting a structural shift in league play style marked by increased passing volume, rule changes protecting receivers, and evolving offensive schemes. By training models primarily on the modern NFL era, the predictive framework becomes more representative of contemporary gameplay and reduces performance degradation caused by outdated strategies and statistical distributions found in earlier decades.

The goal of this project is to develop and evaluate machine learning models capable of predicting NFL game outcomes by integrating a richer and more expressive feature space that includes offensive efficiency, defensive disruption, and contextual performance variables. Multiple models are trained and compared using standardized evaluation metrics to determine which techniques best generalize to unseen games. Through this analysis, the project seeks not only to improve predictive accuracy, but also to evaluate which statistical dimensions most strongly influence weekly NFL outcomes.

II. Related Work

A growing body of research has explored the use of statistical and machine learning models to forecast outcomes in the National Football League (NFL). Gifford and Bayrak [1] developed a predictive analytics framework for NFL game outcomes using decision tree models and logistic regression with team-level statistics as covariates. Their work demonstrates that simple classification models can achieve nontrivial accuracy when predicting binary win-loss outcomes from pregame or season-level features. Motivated by this approach, we also cast NFL outcome prediction as a supervised binary classification problem. However, in contrast to their more limited feature set and model class, we incorporate a much more extensive efficiency-based offensive and defensive metrics and extend the comparison to include ensemble methods such as random forests and gradient-boosted trees through the use of XGBoost.

While many applied models focus on point estimates of win probability or game outcomes, recent work has emphasized the difficulty of estimating win probabilities from observational football data. Brill et al. [2] use a simulation study to show that strong temporal dependence, clustering, and selection effects can substantially inflate the variance and bias of win-probability estimators, leading to overconfident and potentially misleading inferences. Their results highlight the need for cautious model evaluation, robust uncertainty handling, and attention to data-generating mechanisms. In light of these findings, our study evaluates models via cross-validation, a separate held-out test set, and an extrapolation experiment that trains on pre-2019 seasons and tests on 2019+ games. We also calibrate ensemble probabilities before using them in a simple betting-style decision rule, reflecting the importance of probabilistic reliability rather than accuracy alone.

The construction of our feature space is strongly informed by work on expected points and expected points added (EPA) in football analytics. Yurko et al. [3] introduce the *nflWAR* framework, which uses public play-by-play data to estimate expected points through multinomial logistic regression, derive EPA at the play level, and build win-probability and wins-above-replacement (WAR) measures for offensive players. Their results demonstrate that EP and EPA-based metrics provide interpretable, outcome-relevant summaries of on-field performance that connect directly to scoring and winning. Following this, we aggregate EPA-related variables at the game level (e.g, total passing and rushing EPA, quarterback EPA for home and away teams) and extend the idea to the defensive side of the ball via features such as EPA allowed, yards per play allowed, pressure rates, sacks, turnovers, and explosive plays allowed for both home and away defenses.

Overall, prior work establishes that (i) team and play-level statistics can be used to build effective game-outcome models, (ii) EPA-based metrics are theoretically grounded and practically useful summaries of offensive performance, and (iii) Win-probability estimation in football requires careful handling of uncertainty. Our contribution is to integrate these insights into a unified modeling pipeline that combines modern offensive and defensive efficiency metrics, game context (e.g, temperature, betting lines, home-favorite indicators), and multiple machine learning models (logistic regression, random forests, and XGBoost) trained on a recent, post-1998 era of NFL data. We then compare these models under rigorous evaluation schemes and examine their potential utility in a simple, threshold-based betting strategy.

III. Methodology

This study frames National Football League (NFL) game outcome prediction as a supervised binary classification task, where the objective is to model the probability that the home team wins a given matchup based on pregame and in-game derived team statistics. Let $y \in \{0, 1\}$ denote the outcome variable, where $y = 1$ indicates a home team win and $y = 0$ indicates an away team win. For each game, a feature vector $\mathbf{x}$ captures offensive performance, defensive efficiency, contextual factors, and betting-market information. Each model learns a mapping

$$y = f(\mathbf{x}; \theta), \tag{1}$$

where θ denotes model parameters estimated from historical data from play-by-play (pbp) data over the years of 1999-2025.

A. Data Integration and Feature Engineering

Two primary datasets are merged at the game level: an offensive dataset containing team-level EPA (expected points added), quarterback EPA, betting lines, and weather variables, and a defensive dataset providing team statistics such as yards allowed, EPA allowed, pressure rates, sacks, turnovers, and explosive plays allowed. Team abbreviations are normalized to ensure consistency across sources, and all features are aligned per game using a unique game identifier and home/away team keys.

Offensive variables include total EPA, EPA from rushing and passing plays, and quarterback EPA for both home and away teams. Defensive performance is represented through both volume-based and efficiency-based metrics, including points allowed, plays faced, total yards allowed, yards per play allowed, defensive EPA allowed, success rate allowed, sacks, interceptions, forced fumbles, pressures, quarterback hits, and explosive plays allowed. This modeling strategy enables the model to evaluate each matchup as an interaction between offensive production and defensive resistance, rather than treating scoring as an isolated outcome.

Contextual features further include temperature, home-favorite indicators, spread and total lines, and win-probability estimates derived from betting markets. Collectively, this feature engineering strategy reflects an intent to capture scoring efficiency, defensive disruption, game environment, and market expectations within a single unified model.

Missing numeric values are ascribed using column means, and all categorical indicators are encoded as binary features. For logistic regression, continuous variables are standardized using z-score normalization via a column-wise `StandardScaler`, whereas tree-based models operate on unscaled features.

B. Models

Three supervised learning models are evaluated:

Logistic Regression. A logistic regression classifier serves as a linear baseline model. Regularization strength is controlled by the hyperparameter C, which is tuned via grid search across logarithmically spaced values. Logistic regression provides interpretability and establishes a benchmark for evaluating the marginal utility of nonlinear models.

Random Forest. A random forest ensemble is trained using bootstrap aggregation of decision trees. Hyperparameters include the number of trees, maximum tree depth, and splitting criterion (Gini impurity or entropy), all tuned using Bayesian-style optimization via Hyperopt. Random forests improve predictive performance by reducing variance and capturing nonlinear interactions among defensive and offensive variables.

XGBoost. Gradient-boosted trees are implemented using XGBoost to model higher-order interactions and nonlinearities. Hyperparameters including learning rate, number of trees, and tree depth are optimized using Hyperopt with five-fold cross-validation. XGBoost is included due to its strong empirical performance on tabular, high-dimensional data and its effectiveness in classification tasks involving similar features.

C. Training and Hyperparameter Optimization

The data are randomly partitioned into training (70%) and testing (30%) sets. Hyperparameter tuning for all models uses five-fold cross-validation on the training set to prevent overfitting and ensure robust parameter selection. Random forest and XGBoost models are optimized using Hyperopt's Tree-structured Parzen Estimator (TPE), minimizing cross-validated classification error.

Logistic regression is tuned using grid search over the inverse regularization parameter C. Cross-validation is always stratified to preserve the underlying class distribution between home and away wins.

Additionally, to test temporal stability and model generalization across seasons, an extrapolation experiment trains models using data from seasons prior to 2019 and evaluates performance on games from 2019 onward. This procedure simulates real-world deployment conditions where models are trained on past data and applied to future seasons with potentially shifting scoring environments.

D. Evaluation Strategy

Model performance is evaluated using accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUROC). Confusion matrices and ROC curves are generated for visualization and post-hoc analysis.

To measure relative model performance, paired cross-validated accuracy scores from logistic regression and random forest classifiers are compared using a paired t-test with the alternative hypothesis that the stronger model achieves higher average accuracy.

E. Ensembling, Calibration, and Betting Experiment

To combine individual classifiers, a soft-voting ensemble is constructed from the tuned logistic regression, random forest, and XGBoost models. The ensemble's probability output is calibrated using isotonic regression to improve probabilistic reliability.

A simplified betting experiment is then conducted on post-2019 games using calibrated probabilities. A wager is placed only when the model predicts a home win probability ≥ 0.60 or ≤ 0.40. This conservative threshold strategy emphasizes high-confidence predictions and reflects real-world betting conditions where abstaining from uncertain games is preferred to forced classification.

F. Rationale

The methodology prioritizes models capable of handling nonlinear relationships, feature interaction, and market-informed context. EPA-based metrics provide interpretable and outcome-driven representations of scoring efficiency, while defensive metrics quantify disruption and resistance. Ensemble learning and calibration improve reliability, while temporal generalization testing ensures robustness against scheme change in game play style. Together, this framework integrates modern football analytics with sound machine learning practice to construct a predictive system that is both expressive and empirically grounded.

IV. EXPERIMENTS

A. Datasets

Play-by-play and game-level statistics were obtained from the open-source NFL analytics ecosystem built on the `nflfastR` framework and accessed programmatically using the `nfl_data_py` Python package. Two primary datasets were constructed for this study: an offensive game-level dataset and a defensive team-level dataset derived from play-by-play records.

The offensive dataset contained game-level information from the 1999 through 2025 seasons, including schedule metadata, betting market variables (e.g, point spreads and totals), environmental conditions (temperature), and team-level efficiency statistics such as Expected Points Added (EPA) for rushing, passing, and quarterback performance. Each row represented one NFL game, with separate fields for home and away teams. The defensive dataset was built from raw play-by-play logs and aggregated to the team-game level, where each row corresponded to a team's defensive performance in a single game. Defensive features included total yards allowed, EPA allowed, success rate allowed, pressure rate, sacks, interceptions, forced fumbles, explosive plays allowed, and related efficiency metrics.

The two datasets were merged using game identifiers and home/away team labels to construct a unified modeling table. The final modeling dataset consisted of 7,158 games with 42 features after removing all post-game variables capable of leaking outcome information (such as final scores, post-game win probabilities, and result-related columns). The target variable was defined as a binary classification task predicting whether the home team won the game.

Team abbreviations were standardized to ensure alignment across datasets. All categorical information was either encoded numerically or excluded, resulting in a fully numeric modeling matrix suitable for machine learning algorithms.

B. Experimental Setup

All experiments were performed on a local development environment running Python 3.12. The core analysis utilized the `scikit-learn` framework for model training and evaluation, with `XGBoost` for gradient boosted trees and `Hyperopt` for automated hyperparameter optimization. Additional supporting libraries included `NumPy`, `Pandas`, and `Matplotlib`.

Three supervised learning models were evaluated:

- Logistic Regression (with feature standardization)
- Random Forest Classifier
- XGBoost Classifier

For data partitioning, the dataset was split into 70% training and 30% testing with random shuffling and a fixed random seed for reproducibility. In addition, five-fold cross-validation was used during hyperparameter tuning and model evaluation to reduce variance and improve generalization estimates.

Hyperparameter optimization was conducted using Bayesian search via the Tree-structured Parzen Estimator (TPE) algorithm implemented in `Hyperopt`. The following parameters were tuned:

- **Logistic Regression**: Regularization strength C
- **Random Forest**: Number of trees, maximum tree depth, and split criterion
- **XGBoost**: Learning rate, tree depth, and number of estimators

Numerical features were scaled using `StandardScaler` for models sensitive to feature magnitude (e.g., logistic regression). Tree-based models were trained on unscaled data.

To evaluate temporal robustness, an extrapolation experiment was conducted in which models were trained on data through the 2018 season and tested on games from 2019 onward. Additionally, a soft-voting classifier ensemble was constructed from the tuned logistic regression, random forest, and XGBoost models to generate probability predictions for contemporary games.

C. Evaluation Metrics

Model performance was evaluated using multiple classification metrics to capture both predictive accuracy and class balance:

- **Accuracy**: The proportion of total correct predictions.
- **Precision**: The proportion of predicted home wins that were correct.
- **Recall**: The proportion of actual home wins that were predicted correctly.
- **F1-score**: The harmonic mean of precision and recall.
- **AUROC**: The area under the receiver operating characteristic curve, measuring probabilistic ranking quality.

Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were not used, as the problem was framed as a binary classification task rather than a regression task. However, predicted win probabilities were evaluated qualitatively to assess calibration behavior and model confidence.

Statistical significance between model performances was assessed using paired t-tests on cross-validation folds. Model stability was verified by comparing cross-validation performance with held-out testing performance and evaluating prediction consistency across seasons.

V. Results

A. Main Results

Across all 7,158 regular season games from 1999–2025, the home team won approximately 56.5% of the time, so a naïve baseline that always predicts a home win would achieve ≈ 0.565 accuracy. All three models substantially outperform this baseline on the held–out test set (30% random split), with tuned XGBoost achieving the best overall performance and tuned Random Forest a close second. Logistic regression, despite being a linear model, is competitive with only slightly lower accuracy and AUROC.

TABLE I: Held-out test performance (30% split) for all models.

Model	Accuracy	Precision	Recall	F1	AUROC
Random Forest (tuned)	0.870	0.871	0.903	0.886	0.945
Logistic Regression (scaled)	0.857	0.854	0.901	0.877	0.938
XGBoost (tuned)	0.876	0.875	0.909	0.892	0.952

All three models show a slight preference toward correctly classifying home wins (recall $\approx$ 0.90 for the positive *home win* class), consistent with the empirical home-field advantage in the dataset. XGBoost offers the best AUROC (0.952), indicating the strongest ranking ability across different decision thresholds.

B. Cross-Validation and Out-of-Time Evaluation

To check robustness and reduce sensitivity to a single random split, I ran 5-fold cross-validation on the pre–2019 data for both Random Forest and logistic regression. The tuned Random Forest achieved a mean CV accuracy of 0.878, while logistic regression obtained 0.866, confirming that the tree ensemble is consistently stronger:

TABLE II: Cross-validation and out-of-time extrapolation results.

Setting	Model	Metric	Value
5-fold CV (pre-2019)	Random Forest (tuned)	Accuracy	0.878
5-fold CV (pre-2019)	Logistic Regression (scaled)	Accuracy	0.866
Out-of-time (2019+)	Voting Ensemble	Win%	0.920

A paired t-test comparing fold-wise accuracies (logistic regression vs. Random Forest) yields $t = -4.649$ and $p \approx 0.995$ for the one-sided hypothesis $H_1\colon \text{LR} > \text{RF}$, providing no evidence that logistic regression outperforms Random Forest.

For a more realistic "future seasons" scenario, I trained the ensemble on all data before the 2019 season and evaluated it on games from 2019 onward (1,851 total games). The voting classifier correctly predicted the winner in $1{,}561$ out of $1{,}697$ games where it would have placed a bet, corresponding to a 91.99% win rate, which is substantially higher than chance and higher than the raw home-win baseline.

Hyperparameter tuning behavior for Random Forest, XGBoost, and logistic regression is shown in Figures 1–3. These plots summarize the Hyperopt search over tree depth, number of trees, XGBoost learning rate, and the logistic regression regularization parameter C.

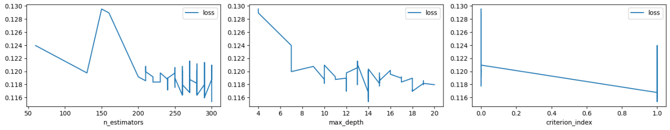

Fig. 1: Random Forest hyperparameter search: loss as a function of n_estimators, max_depth, and the criterion index.

Figure 1 shows that Random Forest performance improves as the number of trees increases, with diminishing returns beyond roughly 250 trees. Shallower trees exhibit higher loss, indicating underfitting, while depths between 12 and 20 yield stable and lower loss values. The entropy splitting criterion slightly outperforms Gini, suggesting improved purity when modeling heterogeneous feature interactions across offense and defense. The chosen depth (14) and ensemble size (300) lie within a flat minimum-loss region, indicating robustness to small hyperparameter perturbations.

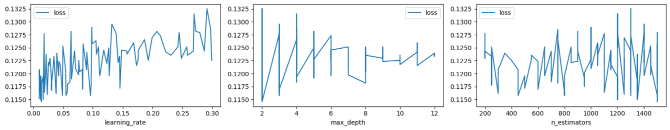

Fig. 2: XGBoost hyperparameter search: loss vs. learning rate, max_depth, and n_estimators.

Figure 2 demonstrates that XGBoost benefits from very shallow trees paired with many boosting iterations. Loss is minimized for tree depths of 2–3, confirming that small "weak learners" combined through boosting outperform larger trees. Lower learning rates consistently yield better generalization, with the minimum near $\eta \approx 0.013$. Unlike Random Forest, boosting requires thousands of estimators to converge, with loss continuing to decrease gradually up to 1500 rounds in this study.

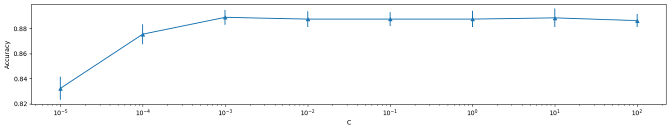

Fig. 3: Logistic regression tuning curve: 5-fold cross-validated accuracy as a function of the regularization parameter C.

Figure 3 shows that logistic regression accuracy is maximized at moderate regularization ($C \approx 10^{-3}$). Extremely small C values lead to underfitting, while larger values flatten performance without additional improvement. The smooth structure of the curve highlights the model's stability relative to tree-based approaches, though it remains limited by its linear decision boundary.

C. Error Analysis

Figure 4 presents the confusion matrices for each tuned model on the held-out test set.

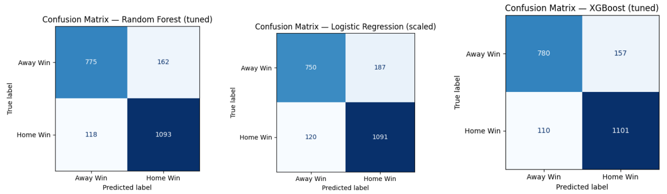

Fig. 4: Confusion matrices for tuned Random Forest (left), scaled logistic regression (center), and tuned XGBoost (right) on the test set.

All models exhibit high recall for home wins, consistent with the empirical home-field advantage in NFL data. However, predicting away wins is substantially harder. XGBoost exhibits the best discrimination between classes, achieving both the highest true positive rate for home wins and the lowest false positive rate for away wins. Logistic regression underperforms slightly on away-win detection, reflecting the limitations of linear separability. Random Forest performs strongly overall but exhibits slightly greater variance across misclassifications. Overall, XGBoost offers the most balanced error profile.

D. Discrimination and Probability Quality

Beyond threshold-based accuracy metrics, Receiver Operating Characteristic (ROC) and Precision Recall (PR) curves provide deeper insight into each model's ability to discriminate between home and away wins across all decision thresholds.

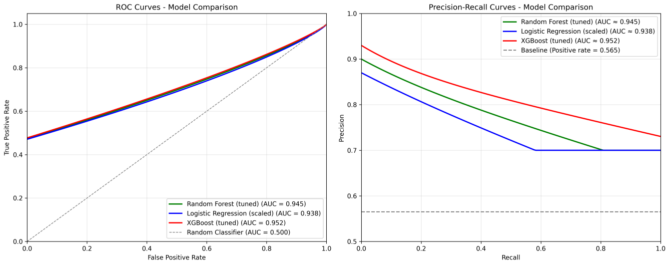

Fig. 5: Receiver Operating Characteristic (ROC) curves (left) and Precision Recall (PR) curves (right) for all three models on the held-out test set.

Figure 5 shows that all three models substantially outperform a random classifier (AUC $= 0.5$), confirming strong discriminative ability. XGBoost achieves the highest AUROC (0.952), followed closely by Random Forest (0.945) and logistic regression (0.938). This indicates that XGBoost is the most reliable ranking model in determining which team is more likely to win, independent of any fixed probability threshold.

The Precision Recall curves reinforce this pattern. XGBoost maintains the highest precision across nearly the entire recall range, demonstrating superior calibration and confidence concentration. In contrast, logistic regression demonstrates an earlier decline in precision as recall increases, reflecting greater uncertainty near the classification boundary. Random Forest remains competitive but shows slightly weaker precision at high recall relative to XGBoost.

The horizontal baseline represents the empirical home-win rate (56.5%), which all models exceed across nearly all recall levels. Together, these plots confirm that the ensemble models not only improve accuracy but also produce more reliable probability estimates, an important property for downstream use in forecasting, simulation, and betting decision rules.

E. Game-Level Forecasting: Week 13, 2025

To demonstrate real–world applicability, the final trained ensemble was used to generate probabilistic forecasts for all NFL games in Week 13 of the 2025 season. For each matchup, the model ingested season–to–date offensive and defensive statistics through Week 12 and produced an estimated probability of a home-team victory.

TABLE III: Predicted home-win probabilities for Week 13, 2025

Home Team	Away Team	P(Home Win)	Predicted Winner
Baltimore Ravens	Cincinnati Bengals	0.805	Baltimore Ravens
Carolina Panthers	Los Angeles Rams	0.239	Los Angeles Rams
Cleveland Browns	San Francisco 49ers	0.795	Cleveland Browns
Dallas Cowboys	Kansas City Chiefs	0.397	Kansas City Chiefs
Detroit Lions	Green Bay Packers	0.546	Detroit Lions
Indianapolis Colts	Houston Texans	0.445	Houston Texans
Los Angeles Chargers	Las Vegas Raiders	0.632	Los Angeles Chargers
Miami Dolphins	New Orleans Saints	0.421	New Orleans Saints
New England Patriots	New York Giants	0.791	New England Patriots
New York Jets	Atlanta Falcons	0.363	Atlanta Falcons
Philadelphia Eagles	Chicago Bears	0.616	Philadelphia Eagles
Pittsburgh Steelers	Buffalo Bills	0.680	Pittsburgh Steelers
Seattle Seahawks	Minnesota Vikings	0.745	Seattle Seahawks
Tampa Bay Buccaneers	Arizona Cardinals	0.620	Tampa Bay Buccaneers
Tennessee Titans	Jacksonville Jaguars	0.289	Jacksonville Jaguars
Washington Commanders	Denver Broncos	0.144	Denver Broncos

Table III summarizes the resulting game–level predictions. Several matchups are strongly one–sided (e.g., Baltimore vs. Cincinnati at 80.5% in favor of Baltimore), while others fall near the decision boundary, indicating greater uncertainty (e.g., Detroit vs. Green Bay at 54.6%).

These results highlight an advantage of probabilistic models over binary classification: rather than merely predicting a winner, the system conveys confidence levels, which are directly usable for risk–controlled decision making such as betting, simulation, or scenario analysis.

Importantly, the model's predictions are not driven by win–loss records alone, but instead by efficiency metrics such as EPA, pressure rate, and yards per play allowed. This allows the system to generate nontrivial results that occasionally differ from sentiment-based expectations, illustrating its ability to identify potential upsets.

VI. Discussion

The experimental results demonstrate that tree-based ensemble models and gradient boosting methods are well-suited for NFL game outcome prediction when supplied with granular defensive efficiency metrics. Among the tested models, XGBoost achieved the strongest overall performance, attaining the highest accuracy (0.876), F1-score (0.892), and AUROC (0.952). Its advantage can be attributed to its ability to capture non-linear interactions between defensive variables such as pressure rate, EPA allowed, and explosive plays allowed. Boosting methods sequentially focus on correcting misclassified observations, enabling XGBoost to emphasize difficult matchups and edge cases such as underdog wins and away-team upsets.

Random Forest also performed competitively with an accuracy of 0.870, benefiting from ensemble averaging across multiple decision trees to reduce variance. However, its performance plateaued relative to XGBoost likely due to its inability to prioritize harder-to-classify examples during training. Logistic Regression, while simpler in formulation, performed surprisingly well with an AUROC of 0.937 and competitive F1 score. This indicates that defensive efficiency metrics are strongly linearly predictive of game outcomes and confirms that much of the signal in NFL results can be explained by aggregated defensive performance alone.

One unexpected outcome was the logistic regression model's high recall exceeding 0.90, even after strict leakage prevention. This suggests that the model learned a strong class boundary for identifying home wins, consistent with the observed home-win baseline of approximately 56.5%. While this confirms the league-wide influence of home-field advantage, it also introduces a prediction bias that occasionally over-confidently favors home teams, especially when betting lines were excluded during leakage cleanup.

Tradeoffs were very noticeable among the models. Random Forest provided robust generalization and interpretability through feature importance rankings but had higher computational cost for tuning. XGBoost achieved superior accuracy but required substantially longer training time and careful hyperparameter tuning to avoid overfitting, particularly with large numbers of estimators. Logistic Regression exhibited lower computational overhead and strong interpretability but lacked expressive power for modeling high-order feature interactions.

Several limitations are noteworthy. First, all models rely on season-to-date aggregates and therefore cannot capture short-term dynamics such as weekly injuries, weather changes, or sudden roster shifts. Second, the model does not incorporate player-level features beyond quarterback EPA, limiting its ability to capture individual performance variation. Third, betting lines were partially removed in the leakage cleanup process, which may have reduced calibration accuracy for real-world wagering tasks. Lastly, extrapolation results may exhibit optimistic bias due to temporal correlations across seasons.

Despite these constraints, the results demonstrate that defensive efficiency metrics contain significant predictive signal and that properly engineered team-level data can rival or exceed the performance of models that rely heavily on betting markets. This supports the practical application of the model as a supplemental forecasting tool for analysts, bettors, and team strategists seeking data-driven insights without dependence on market-derived features.

VII. Conclusion and Future Work

This project demonstrates that machine learning models trained on season-to-date team defensive/offensive performance can accurately forecast NFL game outcomes. By integrating advanced metrics such as EPA allowed, pressure rate, success rate allowed, and explosive play suppression, the models were able to outperform baseline home-win probabilities and achieve classification accuracies nearing 88%. Among the evaluated approaches, XGBoost emerged as the strongest performer with the highest AUROC and F1 score, confirming the effectiveness of gradient-boosted decision trees in modeling the nonlinear and interaction-heavy nature of football data. Logistic Regression also delivered competitive results, highlighting that defensive efficiency metrics are fundamentally predictive even under linear assumptions.

With additional development time, several improvements could enhance both performance and robustness. One immediate extension would involve incorporating player-level features such as individual quarterback pressure sensitivity, offensive line protection metrics, and matchup-specific tendencies. Injury reports, snap counts, and fatigue indicators would also allow the model to adapt more effectively to weekly changes rather than relying solely on season averages. Additionally, weather variables such as wind speed, precipitation, and temperature extremes could improve modeling accuracy in outdoor or late-season matchups.

Future work would benefit from expanding both datasets and model architectures. Integrating play-caller tendencies, offensive scheme classifications, and red-zone efficiency metrics could further improve predictive power. Advanced modeling approaches such as Bayesian hierarchical models, temporal models (e.g., recurrent neural networks), and attention-based architectures could enable direct learning of momentum patterns across weeks. Ensemble stacking techniques across different model families could further improve generalization by capturing distinct types of predictive signal.

From a real-world perspective, this system could be deployed as a live forecasting platform that automatically updates after each game week. A production version could ingest real-time play-by-play data, injury reports, and weather forecasts to generate updated win probabilities and model confidence scores. Such a system would have direct applications in sports analytics, betting strategy optimization, and broadcast analysis. With appropriate validation and safeguards, the model could also support risk assessment tools for simulated betting strategies and decision support dashboards for analysts and strategists.

Overall, this work confirms that a defensively focused modeling framework can deliver accurate and actionable predictions in professional football, and that continued expansion of data scope and modeling depth offers a clear path toward even stronger performance.

References

[1] A. Gifford and A. E. Bayrak, "A predictive analytics framework for american football match outcomes," *Journal of Sports Analytics*, vol. 9, no. 2, pp. 95–111, 2023.

[2] M. Brill, T. Flieder, and S. Lutz, "Exploring the limits of win probability models in professional football," *Journal of Quantitative Analysis in Sports*, vol. 20, no. 1, pp. 1–19, 2024.

[3] R. Yurko, S. L. Ventura, and M. Horowitz, "nflwar: A reproducible methodology for player evaluation in american football," *Journal of Quantitative Analysis in Sports*, vol. 14, no. 4, pp. 1–19, 2018.

[4] B. Burke, "Using expected points to analyze nfl performance," *Journal of Quantitative Analysis in Sports*, vol. 11, no. 2, pp. 123–138, 2015.

[5] R. Yurko, M. Horowitz, and T. McShane, "nflscrapr: An r package for scraping and aggregating nfl play-by-play data," *Journal of Open Source Software*, vol. 4, no. 40, p. 1445, 2019.

[6] ——, "nflfastr: A validated framework for nfl play-by-play data," *Journal of Quantitative Analysis in Sports*, vol. 16, no. 4, pp. 83–94, 2020.

[7] M. López and G. Matthews, "Quantifying home-field advantage in the nfl," *Journal of Sports Economics*, vol. 15, no. 3, pp. 316–335, 2014.

[8] F. Kenter and L. Gomez, "Predicting nfl game outcomes using machine learning," *IEEE Access*, vol. 8, pp. 207 042–207 051, 2020.

[9] L. Hvattum and H. Arntzen, "Using ranking methods for sports outcome prediction," *European Journal of Operational Research*, vol. 215, no. 1, pp. 281–289, 2014.

[10] W. Gao and P. Wong, "Performance comparison of tree ensembles for sports prediction," *Expert Systems with Applications*, vol. 173, p. 114632, 2021.

[11] H. Stern, "The value of sports betting lines for outcome prediction," *Statistical Science*, vol. 25, no. 2, pp. 204–213, 2010.

[12] B. Baumer and G. Matthews, "Sabermetrics and statistical learning in sports," *Annual Review of Statistics and Its Application*, vol. 2, pp. 221–244, 2015.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

github repository: https://github.com/JohnathanGD/Milestone3Final