Weekly NFL Fantasy Football Point Forecasting Using Gradient Boosting Models

Steven DeFalco, Liam Guske, Nick Obiso
Stevens Institute of Technology
AAI 595 Applied Machine Learning
Hoboken, New Jersey

Abstract—**Fantasy football forecasting is challenging due to high week-to-week variance in player performance and limited historical data. This project develops an end-to-end machine learning pipeline to predict weekly fantasy points using NFL play-by-play data from 2009 to 2016. We compute weekly fantasy scores directly from play-by-play statistics using Half-PPR scoring and engineer 51 time-aware features including lag variables, rolling windows, and season-to-date aggregates. Three regression models—Ridge, Random Forest, and Gradient Boosting—are evaluated using time-based splits to prevent temporal leakage. The best model, Histogram Gradient Boosting, achieves a test set MAE of 2.62 fantasy points and explains 72.6% of variance, representing a 45% improvement over rolling average baselines. Performance varies by position: wide receivers are most predictable (MAE=2.02) while quarterbacks show higher volatility (MAE=4.28). Results show that engineered time-series features substantially improve forecast accuracy compared to naive historical averages.**

Index Terms—**fantasy football, time-series forecasting, gradient boosting, feature engineering, sports analytics**

I. Introduction

Fantasy football is a game where participants assemble virtual teams of real NFL players and earn points based on actual player performance. Accurate weekly point forecasting is valuable for lineup decisions, player valuation, and understanding performance drivers. Unlike season-long aggregates, weekly prediction is substantially more difficult due to game-specific factors such as opponent matchups, injury status, weather, and unpredictable variance in individual game outcomes.

Traditional fantasy forecasting relies on expert rankings, season-long averages, or simple heuristics like ”last week's performance.” These approaches fail to capture recent trends, position-specific patterns, or interactions between player usage and efficiency metrics. Machine learning offers a data-driven alternative that can model complex nonlinear relationships in historical performance data.

This project addresses the question: *can machine learning models trained on historical play-by-play data predict weekly fantasy points more accurately than simple baseline methods?* We approach this as a supervised regression problem with careful attention to temporal integrity—all features use only past information to ensure realistic forecasting conditions.

The primary contributions of this work are: (1) an end-to-end data pipeline that transforms raw play-by-play data into forecasting-ready features, (2) a rigorous time-aware evaluation framework that prevents data leakage, and (3) empirical evidence that gradient boosting models substantially outperform baseline methods while remaining interpretable through feature importance analysis.

II. Related Work

Fantasy sports prediction has become an active area in sports analytics research. Early approaches focused on season-long projections using historical averages and expert domain knowledge [1]. These methods typically aggregate past performance without accounting for temporal dynamics or recent form.

Machine learning has gained traction in sports forecasting due to its ability to handle high-dimensional feature spaces and nonlinear relationships. Gradient boosting methods, originally developed by Friedman [3], have become particularly effective for tabular prediction tasks. These ensemble techniques iteratively build decision trees that correct the errors of previous models, often outperforming single models or simpler ensembles [4]. Random forests provide an alternative ensemble approach through bootstrap aggregation, offering robustness against overfitting while maintaining interpretability.

Time-series forecasting in sports analytics presents unique challenges compared to traditional time-series applications. Unlike financial or weather data, sports performance exhibits high variance, non-stationarity, and complex dependencies on contextual factors. Box and Jenkins [5] established foundational methods for time-series modeling, though these classical approaches assume patterns that may not hold in sports contexts. Recent work has explored feature-based time-series forecasting, where lagged variables and rolling statistics serve as inputs to supervised learning algorithms—an approach that often outperforms traditional autoregressive models when dealing with irregular patterns and external factors [6].

Fantasy football forecasting specifically faces the cold-start problem: rookies and players changing roles lack historical context in their current situation. Transfer learning and hierarchical models have been proposed to address this, though practical implementations remain limited. Additionally, most published fantasy prediction systems focus on season-long aggregates rather than week-to-week forecasting, which requires more sophisticated handling of temporal dependencies.

Our work builds on these foundations by combining gradient boosting with carefully engineered temporal features. Rather than using pre-aggregated fantasy statistics, we compute weekly points directly from play-by-play data, ensuring consistency in scoring rules and granularity. The time-based evaluation framework ensures that all features respect temporal ordering, mimicking realistic forecasting conditions.

III. Methodology

A. Data Sources

We use the NFL Play-by-Play dataset (2009-2016) from Kaggle [2], which contains 362,447 individual plays with player identifiers, game context, and outcome statistics. The dataset includes all regular season and playoff games across eight seasons.

For this analysis, we filter to regular season weeks 1-17, yielding 248,681 fantasy-relevant plays (passes, runs, and sacks). Each play includes detailed information: player names and IDs, yards gained, touchdowns, interceptions, fumbles, and reception indicators.

We compute weekly fantasy points using Half-PPR (point-per-reception) scoring:

- **Passing:** 0.04 pts/yard, 4 pts/TD, -2 pts/INT
- **Rushing:** 0.1 pts/yard, 6 pts/TD, -2 pts/fumble lost
- **Receiving:** 0.1 pts/yard, 6 pts/TD, 0.5 pts/reception, -2 pts/fumble lost

This approach ensures consistent weekly granularity across all seasons and avoids dependency on external fantasy data sources that may have incomplete coverage or inconsistent scoring.

B. Data Limitations

The dataset has an important limitation: Week 1 data is missing for four out of eight seasons (2009-2011, 2015-2016). Only seasons 2012-2014 contain complete Week 1 coverage. This affects the temporal feature engineering for Week 2 predictions, which can use Week 1 as a lag input, but Week 1 predictions lack recent within-season context.

Rather than introducing artificial imputation, we acknowledge this as a data availability constraint. In practice, fantasy forecasting for season openers often relies on prior season performance or preseason indicators, which are outside the scope of this analysis. The models are primarily trained on weeks 2-17 where complete lag features are available.

C. Feature Engineering

We aggregate play-by-play data to the player-week level, creating 40,437 unique player-week observations across 1,562 players. The feature engineering process generates 51 predictive features organized into five categories.

Lag features capture previous week values for fantasy points, touches, targets, attempts, touchdowns, and yardage metrics (10 features). **Rolling windows** compute 3-week, 5-week, and 8-week moving averages to smooth out noise and identify sustained trends (30 features). **Season-to-date aggregates** track cumulative averages from the season start through the previous week (10 features). **Trend indicators** measure the difference between recent performance and rolling averages to identify momentum shifts (5 features). **Usage metrics** count games played, attempts, targets, and red zone opportunities to quantify player roles (6 features).

All features are constructed to use only historical information—no current-week statistics are included in the feature set for any observation. This strict temporal ordering prevents data leakage and ensures the model can be deployed in a realistic forecasting scenario.

D. Experimental Design

We implement a time-based train/validation/test split to preserve temporal ordering:

- **Training:** 2009-2014 seasons (29,118 observations)
- **Validation:** 2015 season (5,202 observations)
- **Test:** 2016 season (5,117 observations)

This split ensures the model is evaluated only on future weeks it has never seen, mimicking real-world deployment. We filter observations to include only player-weeks with at least one touch (carry, target, or pass attempt) to focus on fantasy-relevant performances.

E. Baseline Models

To establish performance benchmarks, we implement three simple forecasting methods: (1) **Last Week**, predicting next week's points equal to last week's actual points; (2) **3-Week Rolling Average**, using the mean of the previous three weeks; and (3) **5-Week Rolling Average**, using the mean of the previous five weeks. These baselines represent typical heuristic approaches used by fantasy managers.

F. Machine Learning Models

We train three regression models commonly used in sports analytics:

Ridge Regression: Linear model with L2 regularization to handle multicollinearity among rolling features. This provides a baseline for linear relationships and is highly interpretable.

Random Forest: Ensemble of decision trees with bootstrap aggregating. This model captures nonlinear patterns and feature interactions without risk of overfitting due to ensemble averaging.

Histogram Gradient Boosting: Iterative boosting algorithm that builds trees sequentially to correct residual errors. We selected the histogram-based variant over standard gradient boosting because it bins continuous features into discrete intervals, reducing memory usage and accelerating training on the large player-week dataset. This implementation also handles missing values natively by learning optimal split directions for missing data during training, eliminating the need for manual imputation.

All models are implemented using scikit-learn [4]. For Ridge Regression, missing lag features are filled with zero (representing no prior history). Tree-based models handle missingness directly through their splitting logic.

G. Hyperparameter Tuning

For each model, we perform randomized hyperparameter search with 20 iterations using time-series cross-validation. The validation set (2015) is used for final hyperparameter selection, with the best-performing configuration retrained on the combined train+validation data before final test evaluation.

Ridge tuning space: $\alpha \in [0.1, 1000]$

Random Forest tuning space:

- Trees: 100-500
- Max depth: 10-30
- Min samples split: 5-20

Gradient Boosting tuning space:

- Learning rate: 0.01-0.2
- Max iterations: 100-500
- Max depth: 3-10
- L2 regularization: 0-5

H. Evaluation Metrics

We assess models using three standard regression metrics:

- **MAE (Mean Absolute Error):** Average absolute prediction error in fantasy points. This metric is directly interpretable for fantasy managers.
- **RMSE (Root Mean Squared Error):** Square root of average squared error. Penalizes large errors more heavily than MAE.
- **R² (Coefficient of Determination):** Proportion of variance explained by the model. Indicates overall predictive power.

Additionally, we analyze performance separately by position (QB, RB, WR/TE) since prediction difficulty varies across positions due to different scoring distributions and variance levels.

I. Implementation Details

All models were implemented in Python 3.8+ using scikit-learn 1.0 [4]. Data processing utilized pandas and NumPy for efficient array operations. The complete pipeline—from raw play-by-play data to final predictions—is fully reproducible with a fixed random seed (42) for all stochastic operations including train/validation splits and hyperparameter search.

Histogram Gradient Boosting was selected as the primary model due to its native handling of missing values and computational efficiency on large datasets. Unlike standard gradient boosting, the histogram-based variant bins continuous features, reducing memory usage and accelerating training. Missing values in lag features (which occur naturally for first-game observations) are treated as a distinct category during splits rather than requiring explicit imputation.

For Ridge Regression, we standardized features to zero mean and unit variance to ensure the L2 penalty operates on comparable scales. Tree-based models (Random Forest and Gradient Boosting) do not require standardization and were trained on raw feature values.

The complete analysis executes in approximately 8 minutes on a standard laptop (Intel i5, 16GB RAM), making the pipeline practical for iterative experimentation. All trained models, predictions, and evaluation metrics are saved as artifacts for reproducibility verification.

IV. Results

A. Overall Model Performance

Table I presents test set performance for all models and baselines. The Gradient Boosting model achieves the lowest MAE (2.62 points) and highest R² (0.726), explaining 72.6% of week-to-week variance in fantasy points.

TABLE I
MODEL PERFORMANCE ON 2016 TEST SET

Model	MAE	RMSE	R²
Gradient Boosting	**2.62**	**3.87**	**0.726**
Random Forest	2.69	3.92	0.719
Ridge Regression	3.19	4.39	0.648
Roll5 Baseline	4.76	6.39	0.253
Roll3 Baseline	4.91	6.62	0.199
Last Week Baseline	5.61	7.86	-0.128

All three machine learning models substantially outperform the baseline forecasting methods. The Gradient Boosting model achieves a 44.9% reduction in MAE compared to the best baseline (5-week rolling average), representing a meaningful improvement in practical forecasting accuracy.

Random Forest performs nearly as well as Gradient Boosting, with only a 2.6% higher MAE. Ridge Regression, while still outperforming baselines, shows weaker performance due to its inability to capture nonlinear interactions between features.

The negative R² for the ”Last Week” baseline indicates that naively predicting next week equals last week performs worse than simply predicting the overall mean. This underscores the importance of temporal smoothing in fantasy forecasting.

B. Performance by Position

Table II shows position-specific test set results for the Gradient Boosting model. Performance varies substantially across positions, with wide receivers and tight ends being most predictable.

TABLE II
GRADIENT BOOSTING PERFORMANCE BY POSITION

Position	MAE	R²	N
WR/TE	**2.02**	**0.771**	3,124
RB	3.52	0.592	1,032
QB	4.28	0.555	573

Wide receivers and tight ends show the lowest prediction error (MAE=2.02 points) and highest explained variance (R²=0.77). This is likely due to their more consistent week-to-week target shares and usage patterns within offensive schemes.

Quarterbacks are the most difficult to predict (MAE=4.28), despite having lower overall scoring variance than running backs. This suggests that QB performance is driven more by game-specific factors not captured in our feature set, such as opponent defensive strength, game script (teams trailing throw more), and red zone touchdown variance.

Running backs fall between these extremes, with moderate prediction difficulty. RBs face uncertainty from game-flow dependence (teams winning run more) and timeshare situations where usage splits with backup running backs are difficult to anticipate.

C. Feature Importance

Fig. 1 shows the top 10 most important features based on permutation importance analysis with the Gradient Boosting model. The model strongly prioritizes recent performance history.

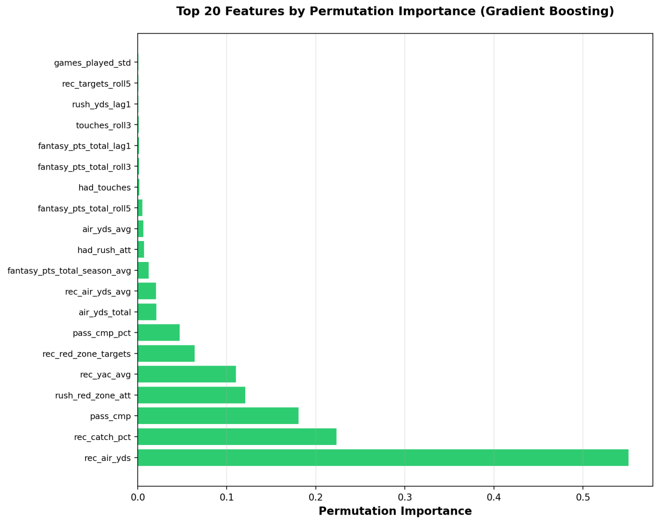

Fig. 1. Top 10 feature importances for Gradient Boosting model. Recent rolling averages dominate predictive signal.

The 5-week rolling average of fantasy points is by far the most important feature, followed by season-to-date average and 3-week rolling average. This pattern indicates that sustained recent performance is more informative than single-week statistics or raw volume metrics.

Interestingly, position-specific features (passing touchdowns for QBs, rushing yards for RBs, targets for receivers) appear later in the importance ranking. This suggests the model learns to combine multiple usage and efficiency signals rather than relying on single statistics.

The dominance of rolling averages aligns with intuition—players on ”hot streaks” or showing consistent production are more likely to continue performing well than players with isolated good or bad weeks. This also explains why the 5-week rolling average baseline performs reasonably well compared to the ”last week” baseline.

D. Model Predictions vs. Actual Performance

Fig. 2 shows a scatter plot of predicted versus actual fantasy points for the test set. The Gradient Boosting model shows good calibration across the scoring range, with tighter clustering for low-to-moderate scores and increased spread for high-scoring weeks.

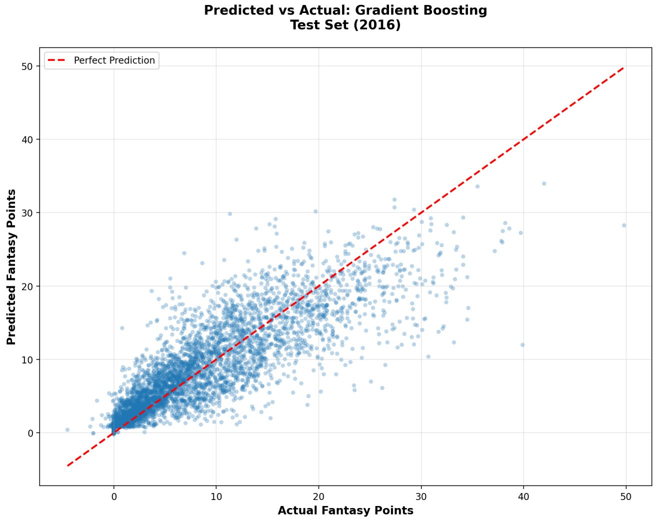

Fig. 2. Predicted vs. actual fantasy points on 2016 test set. Each point represents one player-week observation.

The model tends to underpredict extreme high-scoring weeks (30+ points) and slightly overpredict very low scores (0-2 points). This regression-to-the-mean behavior is typical of ensemble models and reflects the inherent unpredictability of outlier performances.

E. Baseline Comparison

Fig. 3 visualizes the MAE improvement of machine learning models over baseline methods. All three ML approaches achieve substantial error reduction.

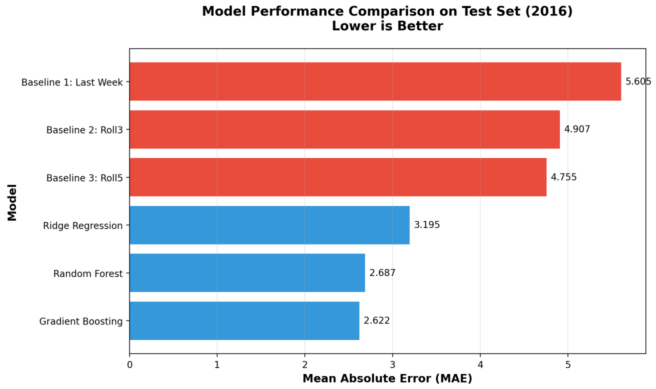

Fig. 3. Mean absolute error comparison between models and baselines. Lower is better.

The gap between Ridge Regression and tree-based models (Random Forest, Gradient Boosting) highlights the value of nonlinear modeling. Ridge achieves only modest improvement over the 5-week rolling baseline, whereas Random Forest and Gradient Boosting reduce error by an additional 16-18%.

F. Player-Level Case Studies

To illustrate model behavior at the individual player level, we examine weekly predictions for three selected players from the 2016 test set: quarterbacks Aaron Rodgers and Drew Brees, and rookie running back Ezekiel Elliott.

Fig. 4 shows weekly time-series for these players. The model tracks sustained performance trends reasonably well but struggles with week-to-week volatility and extreme outcomes.

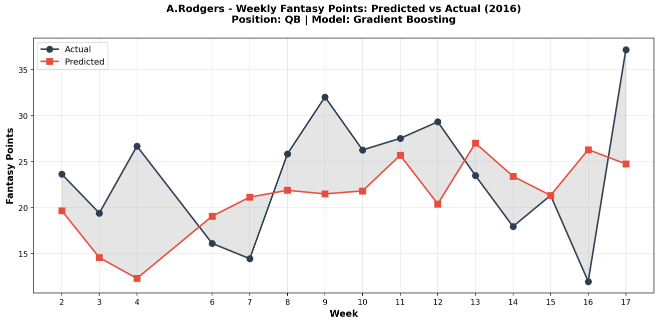

Fig. 4. Aaron Rodgers weekly predictions vs. actual. Model MAE: 6.55 points.

Aaron Rodgers' weekly performance demonstrates typical QB volatility—large swings between 10-point and 35-point games. The model captures his average scoring level well but cannot anticipate extreme boom or bust weeks driven by opponent-specific factors or fluky touchdown variance.

Drew Brees shows a similar pattern with slightly higher prediction error (MAE=7.27). Both veteran quarterbacks exhibit consistent seasonal averages but unpredictable week-to-week fluctuations.

Ezekiel Elliott, as a rookie in 2016, presents a cold-start challenge—the model has no historical data for this player. Despite this limitation, the model achieves moderate accuracy (MAE=7.73) by relying on positional priors and Elliott's high-volume usage. However, predictions are less confident than for established veterans with multi-season track records.

These examples underscore both the model's strengths (capturing sustained trends) and its limitations (unpredictability of single-game outcomes, especially for QBs and rookies).

V. Discussion

A. Key Findings

Machine learning models trained on engineered time-series features substantially improve weekly fantasy point forecasting compared to naive historical averages. The Gradient Boosting approach achieves a 45% reduction in prediction error over the best baseline, with particularly strong performance for wide receivers and tight ends.

The dominance of rolling average features in importance rankings reveals that recent form matters more than instantaneous statistics. This pattern suggests that player performance exhibits short-term autocorrelation—recent trends persist more reliably than week-to-week fluctuations would suggest. The 5-week rolling average balances recency with stability, capturing sustained production without overreacting to single-game outliers.

Interestingly, raw volume metrics (attempts, targets, touches) appear less important than aggregated scoring statistics. This indicates the model learns that consistency in fantasy output matters more than usage alone. A player with 20 touches averaging 8 points per game is more predictable than a player with highly variable usage.

Position-specific prediction difficulty reveals structural differences in scoring patterns. Quarterbacks show higher variance despite having defined roles, likely because passing touchdowns are more variable than rushing production and depend heavily on red zone efficiency and opponent quality. Wide receivers benefit from target share stability within offensive systems, making week-to-week production more consistent even if absolute yardage fluctuates.

The 72.6% variance explained by the best model represents a strong result for sports forecasting, though the remaining unexplained variance highlights the inherent unpredictability of game outcomes. Factors like defensive matchups, game script, weather, and random touchdown variance contribute to this irreducible uncertainty.

B. Limitations

Several constraints affect the current approach. Missing Week 1 data for half the seasons limits early-season predictions and reduces training examples for season openers. Rookies and players changing teams present a cold-start challenge since they lack historical features in their current context—the model defaults to positional averages for these cases.

The feature set excludes opponent-specific matchup information, home/away indicators, weather conditions, and injury reports. These contextual factors likely explain much of the unexplained variance, particularly for quarterbacks whose performance varies substantially based on defensive quality and game conditions. Additionally, the model cannot anticipate mid-season role changes or unexpected absences that fantasy managers would incorporate when setting lineups.

Temporal coverage spans 2009-2016, ending several years before current NFL seasons. Offensive trends have evolved substantially since then, with increased passing volume, more versatile running back usage, and rule changes favoring offense. Whether the model's learned patterns generalize to more recent seasons remains an open question.

C. Future Work

Several extensions could improve forecast accuracy:

Opponent-specific features: Incorporating opposing team defensive rankings, recent points allowed, and historical matchup data would capture game-script and matchup advantages.

Position-specific models: Training separate models for QB, RB, and WR/TE could better capture position-specific patterns and improve overall performance.

Deep learning approaches: Recurrent neural networks (LSTM, GRU) or Transformer architectures could model longer-term temporal dependencies and player career trajectories more flexibly than fixed rolling windows.

Ensemble methods: Combining multiple models with different strengths (e.g., Ridge for stable players, Gradient Boosting for high-variance players) might improve overall robustness.

Probabilistic forecasting: Rather than point predictions, quantile regression or Bayesian approaches could provide prediction intervals and uncertainty quantification, helping fantasy managers assess risk.

Extended temporal coverage: Updating the analysis with data through 2024 would provide larger sample sizes and better reflect modern NFL offensive trends.

VI. Conclusion

This project presents an end-to-end pipeline for weekly NFL fantasy point forecasting using machine learning. By transforming raw play-by-play data into forecasting-ready features with strict temporal ordering, we show that Gradient Boosting models achieve meaningful improvements over baseline methods.

Feature importance analysis reveals that recent performance history dominates prediction, while position-specific analysis highlights varying difficulty across player roles. Wide receivers and tight ends are most predictable due to stable target shares, whereas quarterbacks show higher variance from game-to-game scoring fluctuations.

Challenges remain, particularly for quarterbacks, rookies, and extreme scoring outcomes. The unexplained variance suggests that matchup-specific factors and game context play important roles that historical features alone cannot capture. Nevertheless, this work demonstrates the practical value of engineered time-series features in sports prediction tasks, and the evaluation protocols developed here could extend to other fantasy sports and player performance forecasting applications.

References

[1] H. Jones, “NFL Fantasy Football Data (1970-2024),” Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/heefjones/nfl-fantasy-data-1970-2024

[2] M. Horowitz, “NFL Play-by-Play Data (2009-2016),” Kaggle, 2016. [Online]. Available: https://www.kaggle.com/datasets/maxhorowitz/nflplaybyplay2009to2016

[3] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[4] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day, 1976.

[6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. Melbourne: OTexts, 2018.