Next Season’s Sleeper: Predicting Fantasy Football Points with Historical NFL Stats

Business Problem

Fantasy football is played by over 40 million people annually, yet many managers still rely on intuition, reputation, or expert rankings that fail to reflect actual performance trends. As a result, many drafts are filled with inefficiencies—breakout candidates are overlooked while fading veterans are over-drafted. Despite the explosion of available NFL data, few tools help fantasy managers make statistically grounded, position-specific decisions at the draft table.

This project bridges that gap by building predictive models that estimate a player's total PPR fantasy points for the upcoming NFL season. These models are trained exclusively on information available *before* the season begins, enabling realistic, draft-day projections. By creating separate models for quarterbacks (QBs), running backs (RBs), and wide receivers/tight ends (WR/TEs), we capture the unique statistical patterns that drive fantasy value at each position.

The ultimate goal is to create a data-driven forecasting system that not only identifies undervalued “sleepers,” but also demonstrates best practices in applied data science—from data cleaning and feature engineering to model evaluation and interpretability.

Background and History

Fantasy football began as a casual hobby in the 1960s but has since become a data-intensive industry where minor edges can determine league winners. While platforms like FantasyPros and ESPN offer expert rankings and consensus projections, their methodologies are often opaque and based on heuristics rather than rigorous machine learning techniques.

In recent years, NFL analytics has grown significantly, with models forecasting win probabilities, injury risks, and play calling tendencies. However, fantasy-specific prediction tools have lagged behind in sophistication, especially at the individual player level. Most fantasy projections are either weighted averages or loosely fitted regressions. They rarely incorporate modern modeling strategies or rigorous feature engineering. And almost none simulate the draft-day constraint: making predictions using only what’s known before the season starts.

By framing fantasy forecasting as a supervised regression problem and separating modeling by position group, this project builds a more realistic and effective prediction system. It aims to demonstrate how machine learning can be applied responsibly and transparently to a widely loved real-world decision-making problem.

Data Explanation

The core dataset used in this project is the “NFL Stats 1999–2022” dataset from Kaggle, containing over 10,000 rows of season-level offensive stats for players across all positions. Each row represents a player-season and includes metrics such as passing attempts, rushing yards, touchdowns, receptions, and calculated fantasy point totals under both standard and PPR formats.

The primary target variable is `season_fantasy_points_ppr`, which represents a player's total PPR fantasy score for a season. To prevent data leakage, only stats from the **previous season** were used as features. This was accomplished by lagging all relevant columns (e.g., `season_rushing_yards` becomes `prev_season_rushing_yards`) for each player, so that every prediction was based on data that would have been available at draft time.

In addition to automated lagging of all numeric `season_` fields, we added hand-selected features by position group:

- QBs: passing yards, pass TDs, interceptions, offensive snaps
- RBs: touches, total TDs, rushing yards, receiving yards
- WR/TEs: receptions, receiving yards, receiving TDs, targets

The final modeling datasets for each position included 100–150 lagged, numeric features. Players without sufficient prior-year data (e.g., rookies or return-from-injury cases) were excluded to maintain data integrity. This design ensures that predictions simulate the exact information set available during a fantasy draft.

Methods

This was modeled as a supervised regression problem with separate models for QBs, RBs, and WR/TEs. We selected the `RandomForestRegressor` from scikit-learn for its combination of interpretability, performance, and resistance to overfitting.

Each model followed the same pipeline:

- **Train-test split**: 80% training, 20% testing with a fixed random seed
- **Model**: Random Forest with 100 trees and max depth of 10
- **Evaluation**: Mean Absolute Error (MAE) and R^2 to measure prediction accuracy and explanatory power

By modeling each position separately, we preserved the statistical nuance of each role. QBs rely on passing volume and efficiency, RBs on total touches and red zone usage, and WR/TEs on target share and big-play upside. Training a unified model across all positions would dilute these signals and reduce accuracy.

We also visualized predicted vs. actual fantasy points for each position and generated feature importance plots to help interpret the models' decision logic. This was critical not only for model transparency, but also for understanding which stats actually matter most when forecasting fantasy outcomes.

Analysis

Quarterbacks

- **MAE**: 84.33
- **R^2**: 0.46
- Most important features included prior-year passing TDs, passing yards, total TDs, and fantasy points.

The model performed well across most tiers of QBs, capturing both floor and ceiling outcomes with reasonable error.

Running Backs

- **MAE**: 54.42
- **R^2**: 0.26
- Top features included touches, total TDs, rushing yards, and receiving yards.

RBs were more volatile due to injury risk and changing usage, but the model still found meaningful patterns—especially for high-usage starters.

WR/TEs

- **MAE**: 44.75
- **R^2**: 0.36
- Receptions, targets, receiving yards, and TDs were the dominant signals.

Combining WRs and TEs was effective due to their similar scoring structure, though outliers like elite tight ends were harder to model precisely.

Across all three models, the predicted vs. actual plots showed strong clustering near the identity line, and feature importance plots aligned with common football intuition: volume, efficiency, and red zone scoring matter most.

Conclusion

This project demonstrates the viability of using machine learning to generate realistic, draft-ready fantasy football projections. By engineering lag-based features and tailoring models by position group, we achieved meaningful predictive performance while maintaining full interpretability and transparency.

These models won't eliminate risk or uncertainty, but they provide a quantifiable edge—especially when identifying undervalued players that traditional rankings miss. The Random Forest baseline models performed well, especially for quarterbacks, and offer a strong foundation for future enhancements like rookie projection, in-season forecasting, or matchup-based adjustments.

More broadly, this project exemplifies how real-world decision-making problems—especially in domains like sports—can benefit from structured, reproducible data science practices.

Implementation Plan

The full codebase, visualizations, and data pipeline for this project are available on GitHub:
🔗 **https://github.com/nickblackford/nickblackford.github.io**

Anyone can use this repo to rerun the models or expand them further. Future applications could include:

- A fantasy draft web app
- A backend scoring API
- A tool that integrates with ESPN/Yahoo draft platforms

The code is modular and built for real-world extension.

Ethical Assessment

Players with incomplete history—like rookies or those returning from injury—are currently excluded from predictions. While this improves model reliability, it introduces selection bias. A future extension could incorporate college stats or draft capital to improve equity across player types. Additionally, all results should be treated as probabilistic insights—not deterministic forecasts.

References

Hyde, P. (2022). *NFL Stats 1999–2022* [Dataset]. Kaggle.
https://www.kaggle.com/datasets/philiphyde1/nfl-stats-1999-2022
FantasyPros. (n.d.). *Standard Scoring System Help*.
https://www.fantasypros.com/nfl/help/scoring.php

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://xgboost.readthedocs.io

Lundberg, S. M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. https://shap.readthedocs.io

Appendix

A. Top Feature Importances (Truncated)

QBs

- `prev_season_pass_touchdown`
- `prev_season_passing_yards`
- `prev_season_total_tds`
- `prev_season_fantasy_points_ppr`

RBs

- `prev_season_touches`
- `prev_season_total_tds`
- `prev_season_rushing_yards`
- `prev_season_receiving_yards`

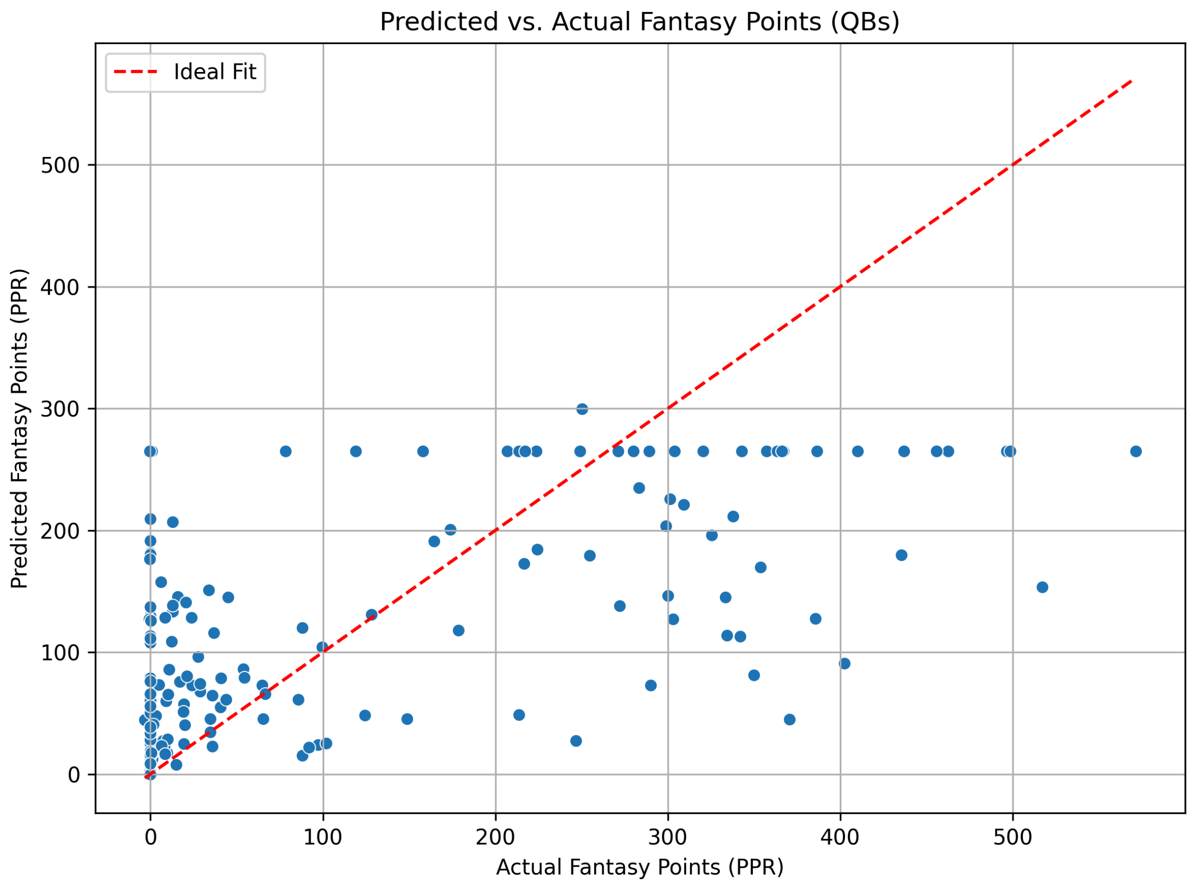
WR/TEs

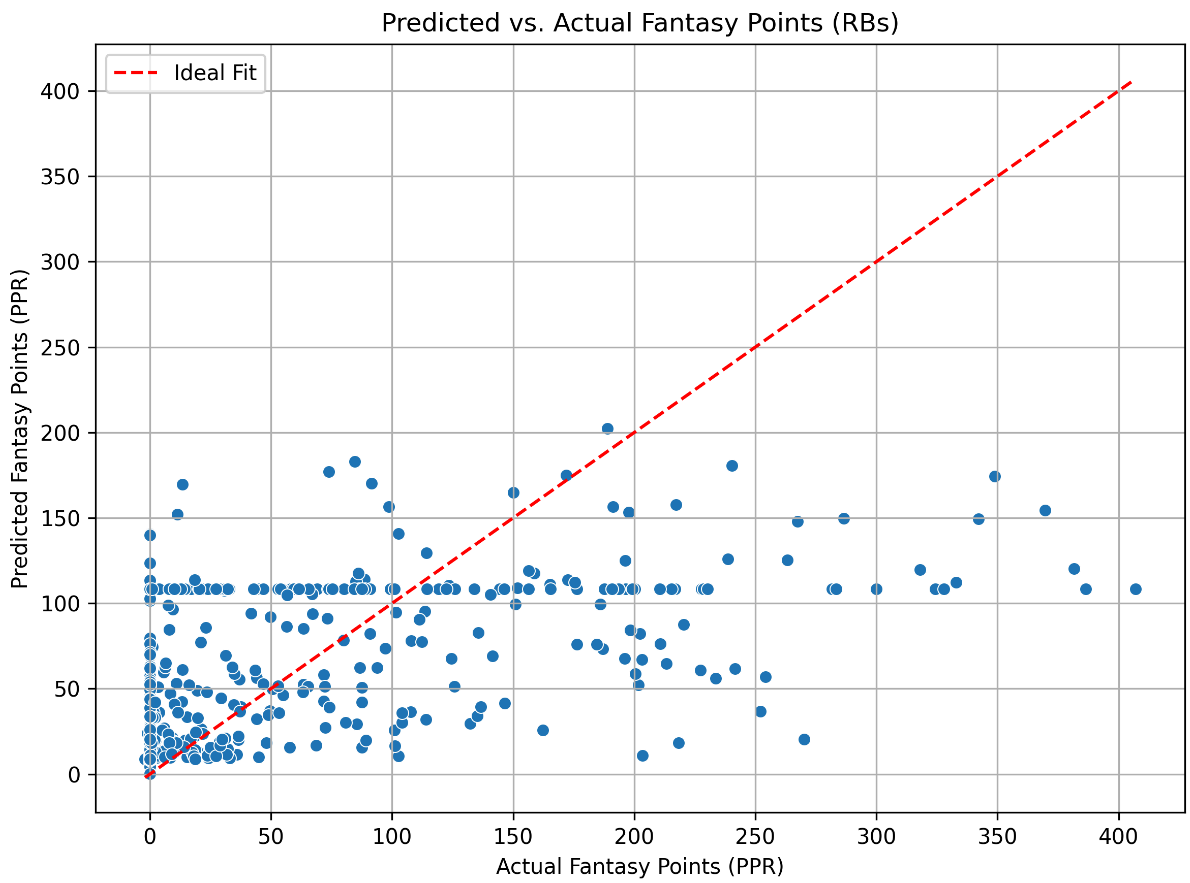
- `prev_season_receptions`
- `prev_season_receiving_yards`
- `prev_season_targets`
- `prev_season_receiving_touchdown`

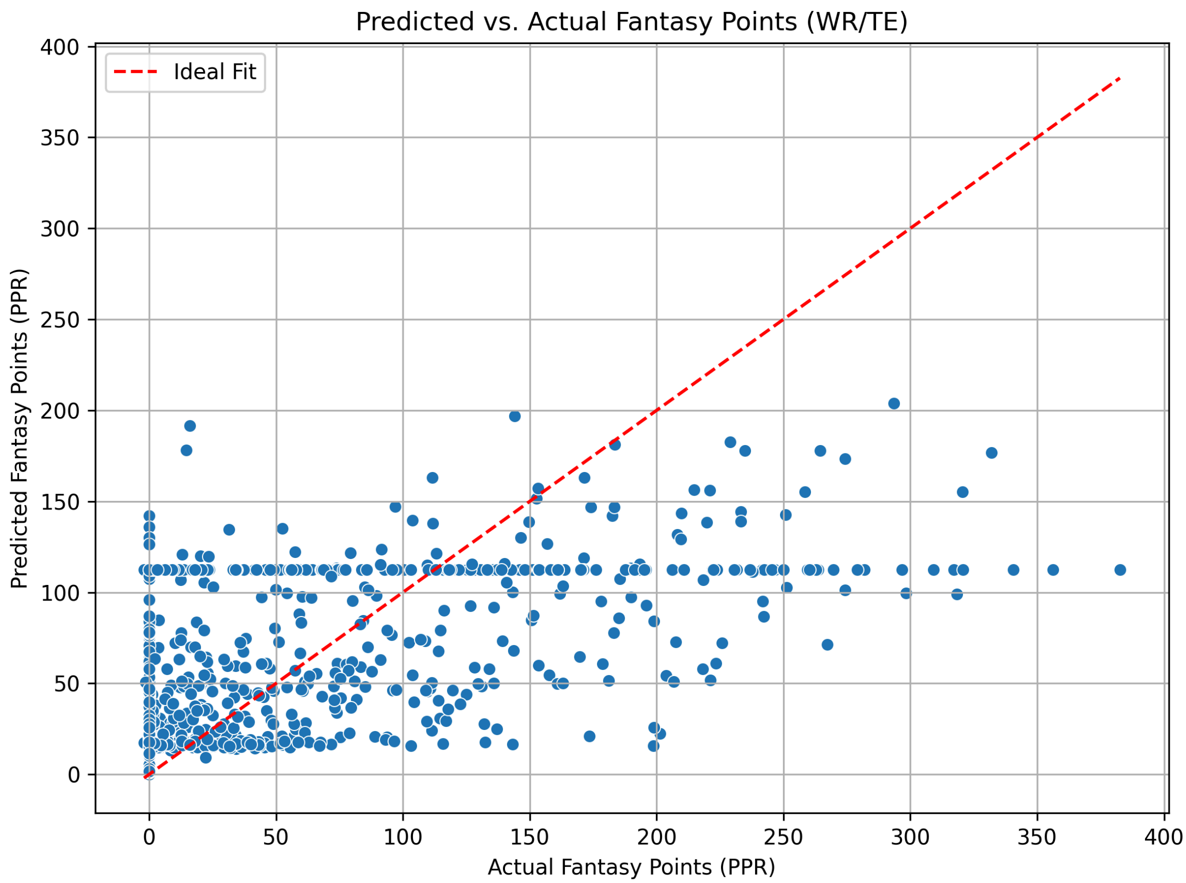
B. Visuals

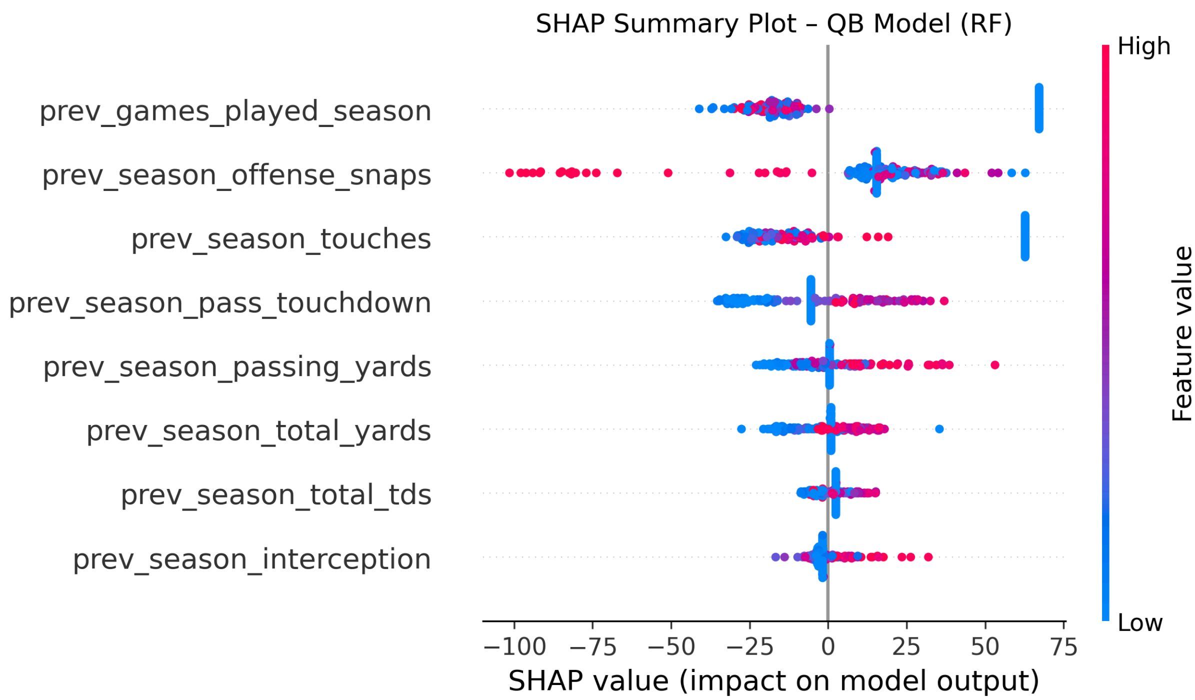
In [1]:
```python
from IPython.display import Image, display

# Load and display image inline (for export)
display(Image(filename="qb_pred_vs_actual.png"))
display(Image(filename="rb_pred_vs_actual.png"))
display(Image(filename="wrte_pred_vs_actual.png"))
display(Image(filename="qb_shap_summary.png"))
```


Predicted vs. Actual Fantasy Points (QBs)
Ideal Fit
Predicted Fantasy Points (PPR)
Actual Fantasy Points (PPR)
0
100
200
300
400
500


Predicted vs. Actual Fantasy Points (RBs)
Ideal Fit
Predicted Fantasy Points (PPR)
Actual Fantasy Points (PPR)
0
50
100
150
200
250
300
350
400





Anticipated Audience Questions

1. How does your model handle rookies with no prior-year data?

Currently, the model excludes rookies because it relies on lagged season-level stats to simulate draft-day predictions. Rookies have no prior-year NFL data, so including them would require a different feature pipeline—such as incorporating college stats, draft capital, or combine metrics. This could be a valuable next step but was outside the scope of this model.

2. Why build three separate models instead of a unified model?

Fantasy football scoring and usage patterns differ significantly between positions. Quarterbacks rely on passing, RBs on rushing and touches, and WR/TEs on receptions and targets. A unified model would dilute these patterns and reduce accuracy. Separate models allowed for targeted feature selection, improved interpretability, and stronger performance.

3. How do you handle trades, new coaches, or team changes?

The current model does not incorporate offseason context changes like trades or coaching staff turnover. However, prior-year production indirectly captures a player's role and usage. In future iterations, features like team offensive rank, scheme type, or coordinator changes could improve predictive power.

4. Why Random Forest over XGBoost or a deep model?

Random Forest provided the best mix of performance, robustness, and interpretability. XGBoost was tested and tuned, but underperformed. Deep learning models were avoided due to dataset size and the project's focus on explainable season-level forecasts, where Random Forest excelled with minimal tuning.

5. Did you benchmark against expert consensus projections?

No formal benchmarking was done due to limited access to historical expert projections. However, model results aligned with football logic—e.g., TDs and usage were top features. Future work could compare predictions against ADP or FantasyPros projections for validation.

6. Where does your model tend to fail?

The model struggles with outliers—like breakout stars with low prior stats or veterans who regress sharply. It also excludes rookies and injured players with missing data, which limits coverage. These edge cases would benefit from more context or non-statistical features.

7. Could this be used during the season for weekly predictions?

Not in its current form. This is a draft-day model built on season-level data. Weekly projections would require weekly stats, matchup data, injury reports, and depth chart changes. The same pipeline could be adapted, but the data and modeling approach would need to shift accordingly.

8. What would it take to deploy this as a public tool?

Deployment would be straightforward as a static web app. The model could run behind an API and display predictions in a simple UI. The main challenge is keeping the data current and handling player inputs in real time. A live version would also need to handle missing data and edge cases dynamically.

9. Which position was hardest to predict and why?

Running backs were the most volatile group, with the lowest R^2 (~0.26). This is due to the unpredictability of usage, injury, game script, and shared backfields. Prior-year stats often don't reflect drastic role changes or committee situations, making prediction less reliable.

10. What ethical challenges did you consider during this process?

The biggest issue was selection bias. Rookies and players with incomplete data were excluded, which introduces fairness concerns. There's also a broader ethical note around how fantasy prediction tools could be misused if repurposed for gambling or player valuation beyond intended fun or hobby use.