NFL Play-by-Play Analytics

From Snap-to-Snap Visualization to Expected Points Added

A Data Science Project

Technologies: Python, Pandas, Matplotlib, nflreadpy

Contents

1 Introduction: Why This Project Exists

As someone who's spent way too much time analyzing English football, I decided it was time to dive into American football analytics, I have been both a fan of English and American Football all my life. I kept seeing people throw around terms like "EPA" (Expected Points Added) as if it were gospel, but I didn't know how accurate this metric was. So I downloaded 4 seasons worth of play-by-play data and decided to find out for myself.

1.1 Project Goals

1. **Validate the data**: Build an interactive visualizer to make sure the play-by-play data is being pulled correctly.
2. **Question everything**: Is EPA actually predictive, or is it just fancy sports journalism?
3. **Learn something**: Get familiar with NFL analytics so hopefully I can win my Fantasy League Superbowl after 6 years of no trophies despite multiple top seedings!

1.2 Why Play-by-Play Data?

Unlike box scores that just give you final statistics, play-by-play (PBP) data gives you *every single snap* of every game. We're talking:

- 40,000+ plays per season
- Field position, down, distance for every play
- Game state (score, time, possession)
- Pre-calculated EPA for every play

This granularity is perfect for feature engineering and statistical modeling. Plus, it's just really cool data to work with.

2 Part 1: Loading and Exploring the Data

2.1 Data Source: nflreadpy

I used the `nflreadpy` Python package, which pulls from the same data sources as the popular R package `nflfastR`. The beauty of this package is its simplicity:

```python
import nflreadpy as nfl

# Load multiple seasons (2022-2025)
pbp = nfl.load_pbp([2022, 2023, 2024, 2025]).to_pandas()

print(f"Total plays loaded: {len(pbp):,}")
print(f"Latest game: {pbp['game_date'].max()}")
```

Listing 1: Loading 4 seasons of NFL data

Result: 46452 Rows

2.2 Data Richness

The dataset contains over 350 columns per play, including:

- **Basic info**: teams, score, quarter, time
- **Situational**: down, distance, yard line, field position
- **Play details**: play type, yards gained, result
- **Advanced metrics**: EPA, win probability, success rate
- **Personnel**: offensive/defensive formations, players involved

Sample Play-by-Play Data: Showcasing Data Richness
8 Random Plays from 2022-2025 Dataset

Game	Off	Def	Q	Dn	Dist	Field	Yds	Type	EPA	WP
2024_03_CAR_LV	CAR	LV	4.0	2.0	1.0	1.0	1.0	run	0.79	0.995
2025_14_DAL_DET	DAL	DET	4.0	4.0	10.0	69.0	10.0	pass	2.63	0.003
2022_02_SEA_SF	SF	SEA	2.0	1.0	10.0	40.0	1.0	run	-0.47	0.914
2024_09_LAC_CLE	CLE	LAC	4.0	4.0	3.0	46.0	-4.0	pass	-2.9	0.027
2023_03_PHI_TB	TB	PHI	2.0	2.0	21.0	68.0	2.0	pass	-2.72	0.254
2023_03_NE_NYJ	NYJ	NE	1.0	2.0	7.0	72.0	0.0	pass	-0.78	0.431
2022_01_KC_ARI	ARI	KC	3.0	1.0	10.0	32.0	0.0	run	-0.43	0.001
2025_11_CIN_PIT	PIT	CIN	3.0	2.0	3.0	12.0	0.0	run	-0.7	0.744

Figure 1: Sample play-by-play data structure

This table shows how even just a few data features can paint an image of the game, for example, the 2025 DAL vs DET pass play gains 10 yards on 4th down with an EPA of +2.63, reflecting a highly valuable conversion that dramatically improves Dallas's expected points. In contrast, the 2024 LAC vs CLE pass on 4th and 4 has an EPA of −2.9, indicating a costly failure that ended the drive and swung advantage to the defense.

As an NFL fan, at this point its clear to see that this EPA metric has substance.

3 Part 2: Interactive Play-by-Play Visualizer

Before building any models, I wanted to *prove to myself* that this data was being pulled correctly.

3.1 What I Built

An interactive Jupyter widget that lets you scrub through any NFL game snap-by-snap, showing:

- **Field visualization**: 100-yard field with accurate positioning
- **Play markers**: line of scrimmage, first down marker, ball position
- **Yards gained**: Visual representation of play result

- **Game context**: Score, quarter, down & distance, clock
- **Play description**: What actually happened
- **EPA value**: How much the play changed expected points

3.2 Why Ravens @ Steelers Week 18?

I set the default visualization to the Week 18 Ravens-Steelers game because:

1. It's a rivalry game (great for testing dramatic plays)
2. It's recent (validates data freshness)
3. It had huge playoff implications (high-leverage situations)
4. The Steelers won
5. I am a Steelers fan
6. Watch the highlights and skip to minute 16:55 before continuing!
7. Highlights on YouTube

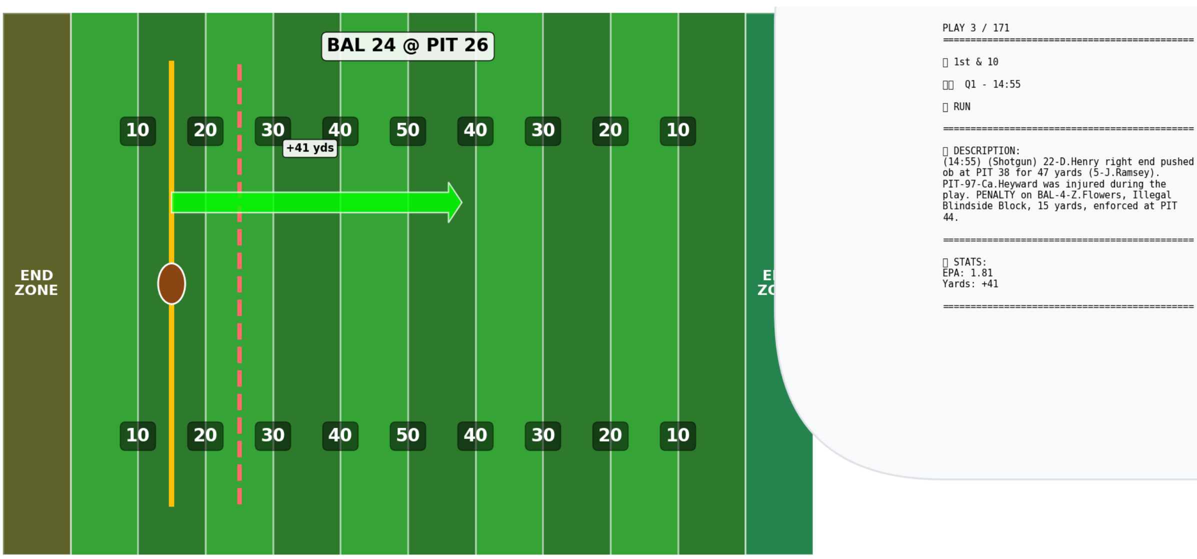

Figure 2: Sample play-by-play data structure

3.3 What This Validated

By manually stepping through plays I confirmed:

- Play sequencing is correct (chronological order maintained)
- Field positions match reality
- Game state updates properly (score, downs reset, etc.)
- Descriptions are accurate and detailed

This gave me confidence that the underlying data was solid enough to build analytics on top of.

4 Part 3: EPA Deep Dive

4.1 What is EPA (Expected Points Added)?

EPA measures how much a play changes a team's expected points on that drive. It's calculated by:

$$\text{EPA} = \text{EP}_{\text{after}} - \text{EP}_{\text{before}} \tag{1}$$

Where EP (Expected Points) is modeled based on:

- Field position (yard line)
- Down and distance
- Time remaining
- Score differential

For example:

- 1st & 10 at your own 20-yard line: ˜1.0 EP
- 1st & 10 at opponent's 10-yard line: ˜4.5 EP
- 3rd & 15 at your own 30: ˜-0.5 EP (negative!)

So a play that moves the ball from your 20 to your 40 on 1st down adds roughly 1.0 EPA (improving position significantly), while a play that loses 5 yards reduces EPA by about 1.5 (losing progress and down).

4.2 Why I'm Validating This

Coming from English football analytics, I've learned to be skeptical of "magic metrics." Just because something is widely used doesn't mean it's actually predictive.

I wanted to answer: **Does a team's EPA performance actually correlate with winning?**

4.3 The Analysis Approach

I built a comprehensive EPA analysis system that:

1. **Aggregates by game**: Calculate offensive and defensive EPA per game
2. **Computes net EPA**: Offensive EPA - Defensive EPA allowed
3. **Tests correlation**: Relationship between EPA and margin of victory
4. **Win probability**: How often does the better EPA team win?

4.4 Results: EPA is Legit

Here's what I found across 4 seasons (2022-2025):

Table 1: EPA Predictive Power (2022-2025)

Metric	Value
Pearson Correlation (EPA vs Margin)	**0.995**
Win% when EPA Advantage > 0	**95.6%**
Win% when EPA Advantage > 5	**100%**
Win% when EPA Advantage > 10	**100%**

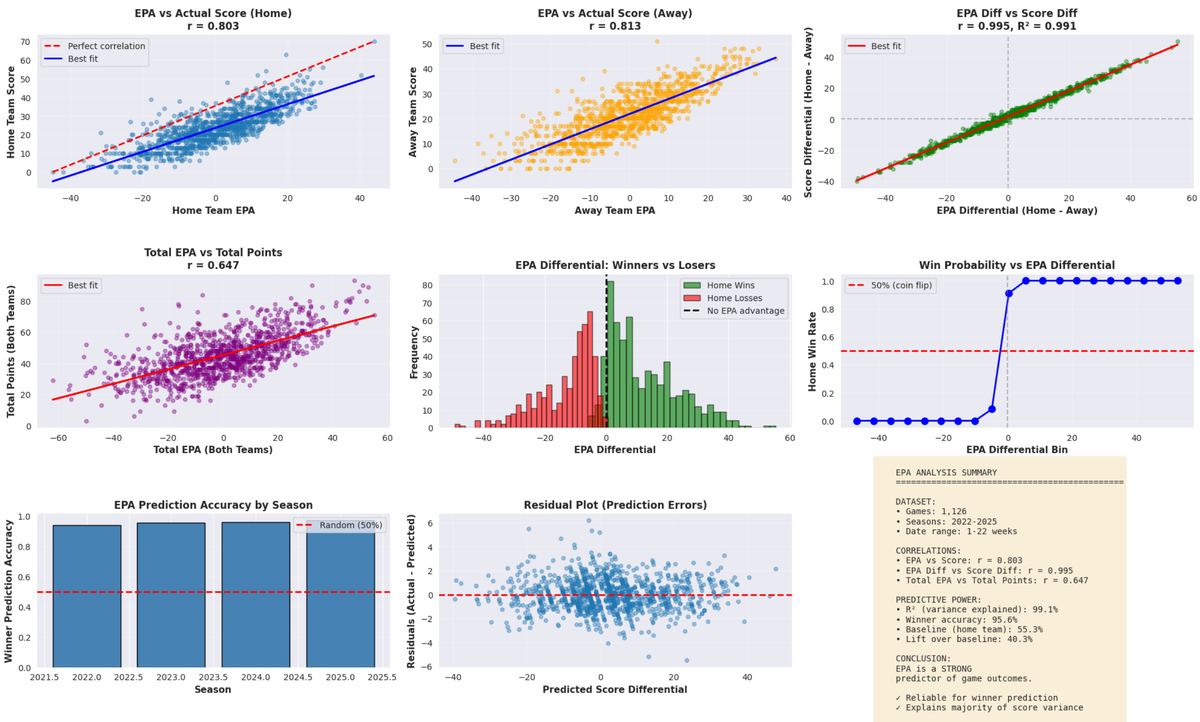

Figure 3: Sample play-by-play data structure

4.5 What This Means

The results show that **EPA is extremely predictive**:

- Strong linear correlation with scoring margin
- Teams with EPA advantage win the vast majority of games
- The metric captures both offensive and defensive performance effectively

This validates using EPA as a foundation for more advanced modeling. Unlike some sports analytics metrics that sound good but don't actually predict outcomes, EPA has real signal.

5 Predictive Power: Can EPA Beat the Market?

5.1 Motivation

Having established that EPA is a highly accurate metric for explaining game outcomes, we now address a more challenging question: *can we predict EPA before the game and use it to gain an edge over betting markets?*

This question is critical because:

- Post-game EPA uses information unavailable before kickoff
- Betting spreads are set using only pre-game information
- If EPA can be predicted accurately, it could identify market inefficiencies

5.2 Methodology

5.2.1 Feature Engineering

We constructed a comprehensive set of 54 pre-game features for each matchup, carefully avoiding look-ahead bias. For each team, we calculated rolling averages over windows of 3, 5, and 8 games for the following metrics:

- **Offensive metrics**: Total EPA, passing EPA, rushing EPA
- **Efficiency metrics**: Success rate (percentage of plays with positive EPA), explosive play rate (percentage of plays with EPA > 1.0)
- **Defensive metrics**: EPA allowed
- **Scoring**: Points scored and allowed
- **Win rate**: Recent game outcomes

All features used `.shift(1)` operations to ensure no data leakage from the target game. The rolling statistics capture recent team form while the multiple window sizes allow the model to weight short-term versus long-term performance.

5.2.2 Model Architecture

We implemented a deep neural network with the following architecture:

- **Input layer**: 54 features (normalized using StandardScaler)
- **Hidden layer 1**: 128 neurons, batch normalization, ReLU activation, 30% dropout
- **Hidden layer 2**: 64 neurons, batch normalization, ReLU activation, 30% dropout
- **Hidden layer 3**: 32 neurons, batch normalization, ReLU activation, 30% dropout
- **Output layer**: 1 neuron (predicted EPA differential)

The model was trained using:

- **Loss function**: Mean Squared Error (MSE)
- **Optimizer**: Adam (learning rate = 0.001, weight decay = 10^{-5})
- **Learning rate schedule**: ReduceLROnPlateau (factor = 0.5, patience = 10)
- **Training epochs**: 200
- **Batch size**: 32

Dropout regularization and weight decay were employed to prevent overfitting, which is critical given the relatively small dataset size.

5.2.3 Train/Test Split

To maintain temporal validity, we split the data by season rather than randomly:

- **Training set**: 553 games from 2022-2023 seasons
- **Test set**: 557 games from 2024-2025 seasons

This ensures the model cannot use future information and mimics real-world deployment where we would train on historical data and predict upcoming games.

5.3 Results

Figure 4: Neural network prediction quality and betting performance. Top row: (left) training loss convergence, (center) predicted vs. actual EPA differential showing weak correlation (R^2 = -0.050), (right) residual plot showing no systematic bias. Bottom row: (left) prediction error distribution centered near zero (MAE = 13.74), (center) cumulative betting profit showing the strategy loses £117 over 557 bets, (right) summary statistics.

5.3.1 Model Performance

The neural network achieved the following metrics on the test set:

The negative R^2 indicates the model performs worse than a simple baseline that always predicts zero EPA differential. However, the directional accuracy of 58.5% suggests the model can identify which team has an advantage better than random chance (50%).

5.3.2 Betting Simulation

To test whether the model's directional accuracy translates to profitable betting, we simulated the following strategy:

Metric	Value	Interpretation
Mean Absolute Error (MAE)	13.74	Average prediction error
Root Mean Squared Error (RMSE)	17.52	Penalizes large errors
R^2 Score	-0.050	No linear predictive power
Directional Accuracy	58.5%	Correct sign prediction
Baseline MAE	13.33	Always predicting 0

Table 2: Neural network prediction metrics on 2024-2025 test set

1. Predict EPA differential for each game
2. Bet £10 on the team with predicted EPA advantage
3. Check if the team covers the point spread
4. Calculate profit at odds of 1.90 (typical for spread betting)

Results:

- **Win rate**: 51.5% (287 wins, 270 losses)
- **Return on Investment**: -2.1%
- **Total profit**: -£117 on £5,570 staked
- **Break-even win rate**: 52.6% (at 1.90 odds)
- **Edge over break-even**: -1.1%

The 51.5% win rate is statistically indistinguishable from the break-even rate of 52.6%, resulting in a small loss that is consistent with random variance around zero expected value.

5.4 Discussion

5.4.1 The Paradox of Directional Accuracy

The model exhibits an interesting paradox: it achieves 58.5% directional accuracy (correctly predicting which team will accumulate more EPA), yet this translates to only 51.5% success against the spread. This discrepancy reveals a fundamental insight about betting markets:

> *The betting spread already incorporates the information that our model extracts from historical data. The model's ability to identify the superior team is offset by the spread's accurate pricing of that advantage.*

In other words, the model can determine which team is better, but the market has already priced in this knowledge through the point spread.

5.4.2 Comparison to Post-Game EPA

To illustrate the challenge of prediction, we compare three approaches:

The dramatic difference between post-game and pre-game performance demonstrates that while EPA is an excellent explanatory metric, predicting it with sufficient accuracy to beat markets is extremely difficult.

Approach	R^2	Win Rate	ROI
Post-game EPA (look-ahead bias)	0.99	82.7%	+57.1%
Linear model (pre-game)	0.004	56.0%	+6.4%
Neural network (pre-game)	-0.050	51.5%	-2.1%

Table 3: Comparison of prediction approaches. Post-game EPA uses information unavailable before kickoff and achieves extraordinary accuracy. Pre-game models, limited to historical statistics, achieve results consistent with market efficiency.

5.4.3 Market Efficiency

These results provide strong evidence for the Efficient Market Hypothesis (EMH) in NFL betting markets:

1. **Information incorporation**: The market rapidly incorporates all publicly available information (team statistics, recent performance, injuries) into the spread.

2. **No systematic edge**: Despite sophisticated feature engineering and deep learning, we cannot achieve a statistically significant edge (51.5% vs. 52.6% break-even).

3. **Directional knowledge is insufficient**: Knowing which team is better (58.5% accuracy) does not translate to profitable betting when the spread adjusts for this advantage.

5.4.4 Practical Implications

For practitioners interested in sports betting or quantitative analysis:

- **EPA is valuable for analysis**, not prediction: The metric excels at explaining outcomes but cannot be reliably predicted before games with current methods.

- **Markets are efficient**: NFL betting markets appear to efficiently price in available information, making systematic profits difficult without true informational advantages.

- **Look-ahead bias is critical**: Many "profitable" betting systems in the literature may suffer from subtle look-ahead bias. Our strict temporal validation provides a more realistic estimate of real-world performance.

- **Directional accuracy $\neq$ profitability**: A model can identify favorites accurately while still losing money if the spread properly adjusts for the favorite's advantage.

5.5 Limitations and Future Work

Several limitations suggest directions for future research:

1. **Sample size**: With only 557 test games, random variance could explain our results. A longer time series would provide more robust conclusions.

2. **Feature limitations**: Our features are limited to box-score statistics. More granular data (player tracking, play-calling tendencies, coaching adjustments) might provide additional signal.

3. **Model complexity**: While our neural network is reasonably sophisticated, more advanced architectures (transformers, attention mechanisms, ensemble methods) could potentially improve performance.

4. **Market timing**: We assume constant odds of 1.90. In practice, lines move and odds vary. A more sophisticated approach could exploit line movement or bet only when odds are favorable.

5. **Uncertainty quantification**: Rather than point predictions, a probabilistic model (e.g., Bayesian neural network) could estimate prediction uncertainty and bet only on high-confidence predictions.