Context-Adjusted Receiver-by-Bucket Target Efficiency for the New York Jets

A play-by-play baseline, uncertainty-aware estimates, and counterfactual target reallocation

Abstract

Raw EPA per target is difficult to interpret because targets occur in different game situations and vary in difficulty. This report presents a context-adjusted analysis of New York Jets pass targets using league play-by-play data. We fit a league baseline model to estimate expected EPA per target from down, distance, field position, time remaining, score differential, and target descriptors, using out-of-fold predictions via game-level cross-fitting to reduce in-sample optimism. We compute residual EPA as observed EPA minus expected EPA and aggregate Jets residuals by receiver and target "bucket" defined by depth (short, intermediate, deep) and direction (left, middle, right). To mitigate small-sample noise, we apply shrinkage toward zero and bootstrap uncertainty intervals. We report a team bucket summary, a receiver-by-bucket recommendation table filtered by uncertainty, and a counterfactual target reallocation estimate intended for decision support and hypothesis generation rather than causal attribution of player skill.

1 Introduction

Target allocation is one of the clearest levers an offense can adjust, but evaluating target effectiveness is confounded by situation and target difficulty. A receiver who is frequently targeted on third-and-long or on low-percentage deep throws may appear inefficient even if their role is inherently difficult. Conversely, a receiver who sees mostly shallow targets in advantageous situations may appear overly efficient.

This work frames receiver usage evaluation as a normalization problem. We estimate a league baseline for expected EPA given observable context and target descriptors, then evaluate Jets targets relative to this baseline via residual EPA. We summarize residual performance by receiver and by target bucket (depth $\times$ direction) to identify usage patterns that appear systematically above or below league expectation in comparable contexts. The deliverables emphasize interpretability, uncertainty, and implementability rather than maximizing predictive accuracy.

2 Data

2.1 Source and unit of analysis

- Unit: **target** (pass attempt with a receiver and valid EPA)
- Scope: league-wide targets for the selected season; Jets subset extracted for analysis

2.2 Scope summary

Season: **2025 regular season**.

- League targets used (after filtering): **16,609**
- Jets targets used (after filtering): **472**
- Jets games represented (unique `game_id`): **17**

2.3 Filtering

Targets are restricted to:

- Regular season plays
- Pass plays with `play_type == "pass"` and `pass_attempt == 1`
- `receiver_player_name` present
- Non-missing `epa`
- Non-missing required fields used in modeling and bucketing: `down`, `ydstogo`, `yardline_100`, `air_yards`, `pass_location`, `pass_length`, `game_seconds_remaining`, `qtr`, `game_id`

2.4 Bucket definition

Each target is assigned to a bucket:

- Depth from air yards:
  - short (≤ 5)
  - intermediate (6–15)
  - deep (≥ 16)
- Direction from pass location: left / middle / right
- Bucket label: `depth_direction` (example: `intermediate_left`)

3 Methods

3.1 Overview

Pipeline:

1. Fit a league baseline model for expected EPA on targets using context + target descriptors
2. Generate out-of-fold expected EPA predictions and compute residual EPA
3. Aggregate Jets residuals by receiver and bucket with shrinkage and bootstrap uncertainty
4. Simulate counterfactual target reallocations and estimate expected EPA gain per game

3.2 Expected EPA baseline model

We model expected EPA per target using features intended to represent situation and target type rather than outcome:

- Numeric context and engineered features: down, yardline_100, game_seconds_remaining, score_differential, qtr, shotgun, no_huddle, ydstogo_cap, air_yards_cap, log_ydstogo, red_zone, late_game, late_down_x_ydstogo
- Categorical target descriptors: pass_location, pass_length

The baseline is a normalization model, not a high-accuracy EPA forecaster.

3.3 Cross-fitting

To avoid computing residuals using in-sample predictions, we use GroupKFold by `game_id`:

- Train the baseline on targets from a subset of games
- Predict expected EPA on held-out games
- Repeat across folds so every target receives an out-of-fold prediction

3.4 Residual definition

For each target i:

$$\texttt{epa\_resid}_i = \texttt{epa}_i - \texttt{exp\_epa\_oof}_i$$

Positive residual means above baseline expectation in that context; negative residual means below baseline expectation.

3.5 Shrinkage

For Jets targets, compute mean residual by receiver-bucket cell (r, b) with count $n_{r,b}$ and mean residual $\bar{m}_{r,b}$. Shrink toward zero:

$$\texttt{shrunk\_resid}_{r,b} = \bar{m}_{r,b} \cdot \frac{n_{r,b}}{n_{r,b} + K}$$

where K is a conservatism parameter.

3.6 Uncertainty and recommendation rule

For each receiver-bucket cell, bootstrap the mean residual and compute a 95% interval (`ci_low`, `ci_high`).

Recommendation filter:

- **Increase** only if `shrunk_resid` > 0 and `ci_low` > 0
- **Decrease** only if `shrunk_resid` < 0 and `ci_high` < 0
- Otherwise: uncertain (reportable but not actionable)

3.7 Counterfactual target reallocation

We estimate expected gains from shifting s targets per game:

- Move targets from low `shrunk_resid` pairs to high `shrunk_resid` pairs
- Gain approximation: sum over moved targets of (destination shrunk residual minus source shrunk residual)

This is directional and depends on feasibility assumptions; defensive adaptation and role constraints are not modeled.

4 Results

4.1 Baseline model adequacy

Out-of-fold metrics:

- MAE: 1.131
- R^2: 0.020

Play-level EPA is high variance; modest R^2 is expected. The baseline is used to remove broad context effects, not to predict EPA precisely.

4.2 Jets bucket profile (team-level)

Table 1 summarizes Jets vs league residuals by bucket (depth × direction).

Table 1: Jets vs league residual EPA by target bucket.

Bucket	Jets n	Jets mean resid	League n	League mean resid	Jets minus league
short_middle	55	0.006	1567	-0.175	0.181
short_right	126	0.104	3680	-0.002	0.106
intermediate_right	47	0.064	1729	0.054	0.010
intermediate_middle	21	-0.048	1372	0.140	-0.188
intermediate_left	67	-0.146	1700	0.045	-0.192
deep_right	28	-0.283	1243	-0.076	-0.207
short_left	88	-0.246	3461	-0.024	-0.222
deep_left	31	-0.484	1263	0.006	-0.490
deep_middle	9	-0.878	594	0.157	-1.034

4.3 Receiver-by-bucket heatmap

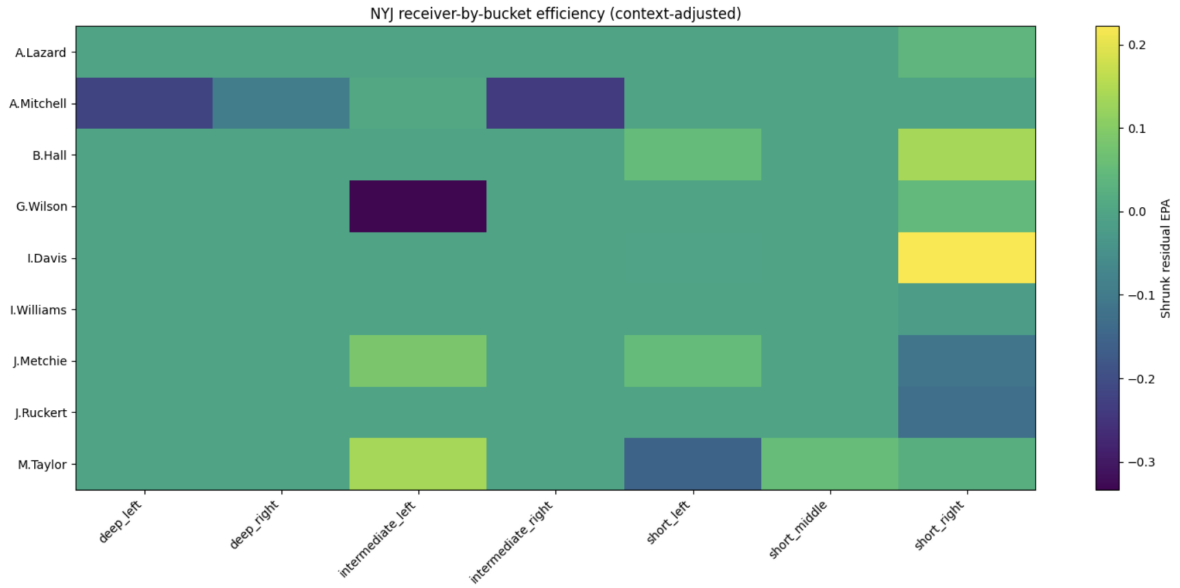

Figure 1: Receiver-by-bucket shrunk residual EPA (Jets). Replace with your generated heatmap file.

Interpret only high-volume, stable cells; sparse receiver-bucket pairs should not be over-weighted.

4.4 Uncertainty-filtered recommendation table

Applying the uncertainty rule (CI does not cross zero), only the following receiver-bucket pairs qualify as actionable based on the exported outputs.

Table 2: Actionable receiver-bucket recommendations after uncertainty filtering.

Receiver	Bucket	n	Mean resid	Shrunk resid	CI low	CI high
M. Taylor	intermediate_left	8	0.573	0.139	0.267	0.906
A. Mitchell	intermediate_right	11	-0.779	-0.238	-1.341	-0.185

4.5 Counterfactual reallocation estimate

We summarize the counterfactual reallocation output numerically (no curve figure included). For shifting $s = 4$ targets per game over 17 Jets games:

- Targets shifted total: 68
- Total EPA gain: 29.196
- EPA gain per game: 1.717

Interpretation: directional and upper-bound-ish, since feasibility and defensive adaptation are not modeled.

5 Discussion

5.1 What the analysis is useful for

- Usage diagnostics: identify receiver-bucket patterns that are efficient relative to baseline
- Film prioritization: focus review on high-volume negative residual pairs
- Gameplanning hypotheses: "lean more into bucket X for receiver Y" as a testable plan

5.2 Main limitations and biases

1. **Target endogeneity.** Targets depend on QB reads, play design, coverage, pressure, and game plan. Residuals are not causal effects.
2. **Omitted variables.** No route, separation, coverage, or pressure timing. Residuals can reflect unobserved conditions rather than receiver quality.
3. **Non-pre-snap conditioning.** Air yards is realized after the play begins; bucket reflects realized target depth, not pre-snap intent.
4. **Within-game dependence.** Plays within a game are correlated; a non-clustered bootstrap can understate uncertainty.
5. **Reallocation feasibility.** Shifting targets between buckets assumes playbook flexibility and ignores defensive adaptation and role constraints.

5.3 Mitigations and robustness checks

- Cluster bootstrap by `game_id` (more conservative intervals)
- Compare two baselines (Ridge vs boosting) and report only stable patterns
- Remove garbage time and check sensitivity
- Add feasibility constraints to reallocation (caps per receiver and bucket)

6 Conclusion

This report provides a reproducible framework to evaluate Jets pass-target usage by receiver and bucket after adjusting for observable context via a league baseline expected EPA model. Residual EPA summaries with shrinkage and uncertainty intervals support more disciplined usage recommendations than raw EPA per target. A counterfactual reallocation estimate provides an interpretable view of potential gains under simplified assumptions. Results should be interpreted conservatively given endogeneity and omitted variables.

A Appendix: Implementation details and parameters

A.1 Run configuration

- Season: `SEASON = 2025`

- Team: `TEAM = "NYJ"`
- Data import: `nfl_data_py.import_pbp_data([SEASON], downcast=True)`

A.2 Filtering rules (exact fields enforced)

From the play-by-play table, the analysis keeps targets satisfying:

- `season == SEASON` and `season_type == "REG"`
- `play_type == "pass"` and `pass_attempt == 1`
- `receiver_player_name` non-missing
- `epa` non-missing
- Drop rows with missing values in:

  {`down, ydstogo, yardline_100, air_yards, pass_location, pass_length, game_seconds_remaining`
- Score differential:
  - If `score_differential` exists, it is used.
  - Otherwise, if `posteam_score` and `defteam_score` exist, set `score_differential = posteam_score - defteam_score`.

A.3 Bucket construction (exact thresholds)

Depth bucket is assigned via:

- short if `air_yards` ≤ 5
- intermediate if $6 \leq$ `air_yards` ≤ 15
- deep if `air_yards` ≥ 16

and then:

$$\texttt{bucket} = \texttt{depth\_bucket} \,||\, \texttt{"\_"} \,||\, \texttt{pass\_location}$$

Targets with missing `depth_bucket` are dropped (after the `air_yards` non-missing filter, this is mainly a guard).

A.4 Feature engineering (exact definitions)

The notebook creates the following engineered features:

- `red_zone` $= \mathbb{1}[\texttt{yardline\_100} \leq 20]$
- `late_game` $= \mathbb{1}[\texttt{game\_seconds\_remaining} \leq 300]$
- `ydstogo_cap` $= \min(\texttt{ydstogo}, 20)$
- `air_yards_cap` $= \texttt{clip}(\texttt{air\_yards}, -5, 40)$
- `log_ydstogo` $= \log(1 + \texttt{ydstogo\_cap})$
- `is_late_down` $= \mathbb{1}[\texttt{down} \in \{3, 4\}]$
- `late_down_x_ydstogo` $= \texttt{is\_late\_down} \cdot \texttt{ydstogo\_cap}$

A.5 Baseline model specification

Inputs:

- Numeric features (`num_feats`):

  ```
  down, yardline_100, game_seconds_remaining, score_differential, qtr,
  shotgun, no_huddle,
  ydstogo_cap, air_yards_cap, log_ydstogo, red_zone, late_game, late_down_x_ydstogo
  ```

- Categorical features (`cat_feats`): `pass_location, pass_length`

Preprocessing and model:

- Numeric pipeline: median imputation (`SimpleImputer(strategy="median")`)
- Categorical pipeline: most-frequent imputation, then one-hot encoding

  `OneHotEncoder(handle_unknown="ignore")`

- Model: `RidgeCV` with

  $$\alpha \in \texttt{logspace}(-3, 3, 25)$$

A.6 Cross-fitting setup

- Splitter: `GroupKFold(n_splits = 5)`
- Group key: `game_id`
- Output: `exp_epa_oof` is the out-of-fold prediction for each target

A.7 Shrinkage and bootstrap parameters

The notebook uses:

- Minimum sample per receiver-bucket cell: `MIN_N = 8`
- Shrinkage strength: `K_SHRINK = 25`
- Bootstrap draws per cell: `B = 500`
- Bootstrap RNG seed: `np.random.default_rng(7)`
- CI extraction: empirical 2.5th and 97.5th percentiles using indices

  $$\lfloor 0.025B \rfloor,\ \lfloor 0.975B \rfloor$$

A.8 Recommendation table selection (not the CI rule itself)

Beyond the CI rule described in the main text, the notebook also sets per-receiver selection caps:

- `TOP_K_PER_RECEIVER = 2`
- `BOT_K_PER_RECEIVER = 1`

These govern how many top and bottom buckets are surfaced per receiver when assembling the candidate recommendation list.

A.9 Reallocation simulation settings

- Default shift rate in code: `shift_per_game = 4`
- Games used when not specified: number of unique Jets `game_id` (17)
- Total shifted targets: `shift_per_game` $\times$ `games` (68)
- Pair definition: `pair = receiver_player_name + " | " + bucket`