BSA Football Spring Presentation

QB Pocket Clutch Ratings

Dataset

nflfastR

- 2021 NFL season, all pass plays
- ~7,000 plays after filtering
- Provides the model's target variable: EPA (Expected Points Added) per play
- Also supplies game context for clutch labeling: win probability, quarter, score differential, time remaining, and down

NFL Big Data Bowl 2023

- 10Hz player tracking for weeks 1-8 of the 2021 season
- x/y coordinates for every player and the ball on every frame
- Paired with PFF scouting tags to identify each player's role on the play (QB, pass blocker, pass rusher)
- Two key frames extracted per play: ball snap and pass release
- All spatial features are computed from these

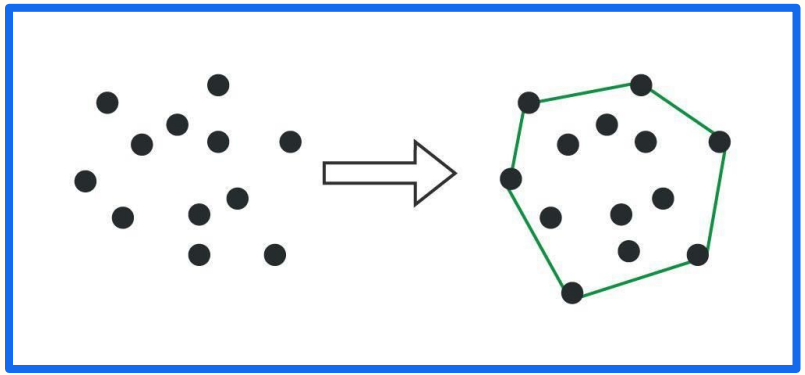
Pipeline Overview

1. (download_pbp.py) — Pull nflfastR PBP, filter to pass plays, assign clutch labels based on Q4/OT win probability and game situation
2. (feature_engineering.py) — Join tracking data to PBP, extract QB, pocket, and rusher features at snap and release frames for each play
3. (model_training.py) — Train XGBoost to predict EPA from spatial features using leave-one-week-out cross-validation (no data leakage)
4. (analysis.py) — Compute optimality (actual − predicted EPA) per play, then compare each QB's clutch plays to their own overall average → clutch rating

Pocket Features

Data/Procedure

- Takes in 2 frames - snap and release
- Uses PFF DataFrame to find the specific play
- Filters through all players to find only ‘Pass Block’ or ‘Pass’
- Uses scipy.spatial (Convex Hull) to calculate area



Output

- Pocket area at release (square yards)
- Pocket collapse rate (square yards/second)
  - (Pocket area @ release - pocker at snap) / Time

Rationale

- Create a feature based on total pocket area
- Take into account how fast/slow the pocket changed

QB Spatial Features

What It Does

- Receives tracking snapshots at the ball snap and pass release frames along with the QB's NFL ID
- Locates the quarterback in both frames using nflId
- Computes QB movement using x-y tracking coordinates
- Extracts speed and body orientation from moment of release
- Calculates the time elapsed between the snap and throw

Returns

1) qb_displacement → float

- Distance traveled by the QB from snap to release (yards)

2) qb_speed_at_release → float

- QB speed at the release frame (yards/second)

3) qb_orientation_at_release → float

- QB body orientation at release (degrees)

4) time_to_throw → float

- Time between snap and release (seconds)

Edge Cases

★ QB missing from snap frame → return NaN values

★ QB missing from release frame → return NaN values

★ Release frame occurs before snap frame → time_to_throw = NaN

Pass Rusher Features

What It Does:

- Receives the tracking snapshot when a pass is released, the QB's NFL ID, and PFF scouting data for the play
- Filters PFF rows where pff_role == "Pass Rush" to identify rushers
- Computes the straight-line distance from each rusher to the QB using (x, y) coordinates
- Sorts rushers by distance and reads the speeds for the 2 nearest to the QB
- Counts how many rushers are within 3 yards of the QB at the time of the release

Returns:

1) nearest_rusher_dist → **float**
   a) Distance (in yards) to closest rusher
2) rusher_1_approach_speed → **float**
   a) Speed of nearest rusher
3) rusher_2_approach_speed → **float**
   a) Speed of 2nd nearest rusher, but returns NaN if it is less than 2 mph
4) rushers_within_3yds → **int**
   a) # of rushers within 3 yards

Edge Cases:

★ No rushers tagged → all values are NaN, rushers_within_3yds == 0
★ < 2 rushers → rusher_2_approach_speed = NaN
★ QB not in frame → raises ValueError (fixed by other model)

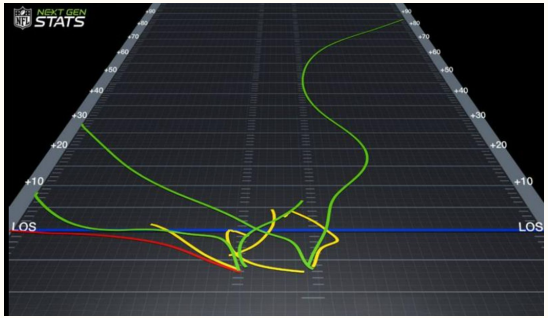
Feature Engineering

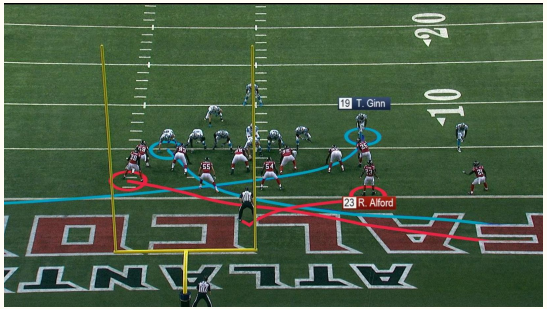
Data/Procedure

- Receives the tracking snapshot when a pass is released, the QB's NFL ID, and PFF scouting data for the play
- Filters PFF rows where pff_role == "Pass Rush" to identify rushers
- Computes the straight-line distance from each rusher to the QB using (x, y) coordinates
- Sorts rushers by distance and reads the speeds for the 2 nearest to the QB
- Counts how many rushers are within 3 yards of the QB at the time of the release

Output

- EPA (target variable)
- QB spatial features
- Pocket geometry features
- Pass rusher features
- Down, distance, score, and time context
- Clutch / non-clutch label





Model

What it Does

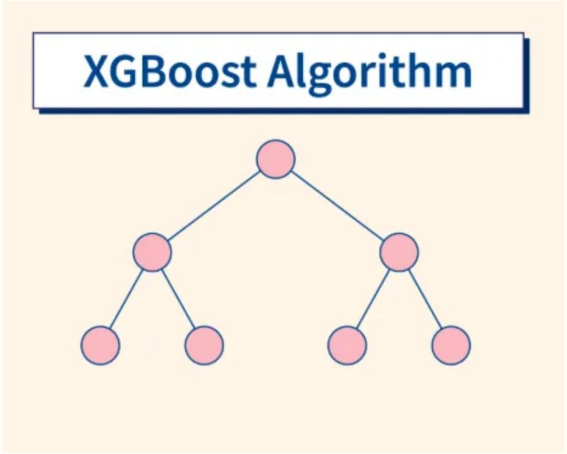
- Trains an XGBoost regressor to predict EPA per play
- Learns relationships between pocket behavior and play success
- Estimates the expected outcome of each play
- Produces predictions without data leakage

Inputs

- QB Features
  - Displacement
  - Speed
  - Orientation
  - Time to throw
- Pocket Features
  - Pocket area
  - Pocket collapse rate
- Pass Rusher Features
  - Nearest rusher distance
  - Rusher speeds
  - Rushers within 3 yards
- Context Features
  - Down
  - Distance
  - Score differential
  - Time remaining

Outputs

- Predicted EPA for every play
- Feature importance rankings
- Play-level prediction table
- Optimality Score: Predicted EPA - Actual EPA



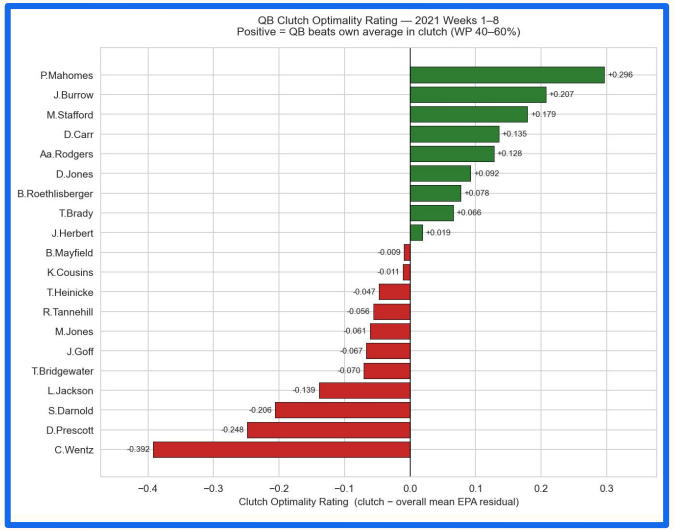
Analysis

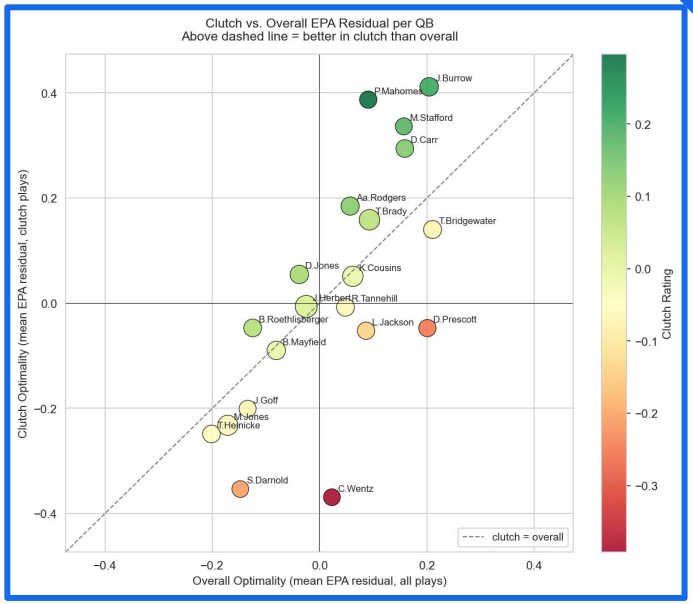
Clutch-optimality = (actual EPA - predicted EPA | clutch situation)

Overall-optimality = (actual EPA - predicted EPA | all plays)

Clutch-rating = Clutch-optimality - Overall-optimality

Clutch-rating is how much better (positive value) or worse (negative value) each QB performs in close games (40-60% WP) vs. how they perform on average.





Filtered for QBs with >= 50 clutch plays and >= 50 non-clutch plays

Potential Drawbacks:

- Model trained only on data from 8 weeks in a single season
- Spatial features are only collected from 2 frames in a play: At the snap and when the QB releases the ball

Future Plans

- Write a paper
- Include more features such as coverage to reduce variability
- Expand rating metric (instead of just EPA) or create our own metric
- Increase number of frames used in pocket, spatial, and pass rusher features
- Train on larger dataset

Thank you!

Any questions?