NFL 2024-2025 Season Data Scraping Documentation

Udayveer Singh Andotra

March 16, 2025

1 Introduction

This document provides an overview of the Python code used to generate an enhanced NFL 2024 regular season dataset, including libraries, code functionality, variables, and team abbreviations. The dataset aggregates play-by-play data, historical head-to-head (H2H) records with ties, team performance metrics, season records with ties, and game context variables, saved as `NFL_DataScrap(2024-2025).csv`.

2 Libraries Used

The following Python libraries are utilized in the script:

- **sys**: Facilitates system-specific operations, used here to install packages via pip.
- **pandas** (as `pd`): Data manipulation and analysis library for handling DataFrames.
- **numpy** (as `np`): Numerical computing library for array operations and mathematical functions.
- **nfl_data_py** (as `nfl`): Specialized library for importing NFL data, including play-by-play and schedules. Source: https://github.com/nflverse/nfl_data_py.

3 Code Description

The script performs the following steps:

1. **Installation**: Upgrades required libraries using `pip`.
2. **Data Loading**: Imports 2024 play-by-play data (`pbp_2024`) and filters for regular season (`season_type == 'REG'`). Loads 2014–2024 historical data for H2H and performance metrics.
3. **Preprocessing**: Computes possession flags (`home_poss`, `away_poss`) and quarter splits (`first_half`, `second_half`, `overtime`).
4. **Aggregation**: Groups 2024 play-by-play data by game, calculating offensive (prefixed `off_`), defensive (prefixed `def_`), and play count statistics.
5. **H2H Calculation**: Computes 10-season H2H records (2014–2023 plus 2024 up to prior week) for each matchup, including ties.
6. **Team Performance**: Adds win/loss/tie streaks, points scored/allowed over the last 3 games, and season records to date with ties.
7. **Game Context**: Incorporates conference/division flags, special game indicators, and schedule data (e.g., spread, total line).
8. **Post-Processing**: Fills NaNs with 0, rounds numeric values to 3 decimal places, and saves the output.

4 Code Algorithm and Functioning

This section outlines the algorithm and detailed functioning of the script in pseudocode and narrative form.

4.1 Algorithm

Input: 2024 NFL season play-by-play data, historical data (2014–2024), 2024 schedule
Output: CSV file with aggregated game-level statistics
Install required libraries (`sys`, `pip`)
Import libraries (`pandas`, `numpy`, `nfl_data_py`)
Load 2024 play-by-play data (`pbp_2024`) from `nfl.import_pbp_data([2024])`
Filter for regular season (`pbp = pbp_2024[pbp_2024['season_type'] == 'REG']`)
Load historical play-by-play data (2014–2024) into `pbp_historical`
Load 2024 schedule data into `schedule_2024`
Preprocess Play-by-Play Data:
Define `home_poss = posteam == home_team`
Define `away_poss = posteam == away_team`
Define `first_half = qtr in [1, 2]`
Define `second_half = qtr in [3, 4]`
Define `overtime = qtr >= 5`
Aggregate Game Data:
Group `pbp` by `[season, week, game_id, home_team, away_team]`
Compute aggregates (e.g., sum offensive yards as `off_yards_gained_h`, mean defensive EPA as `def_epa_per_play_h`, count plays per half)
Calculate `point_diff = total_home_score - total_away_score`
Set `home_win = 1` if `point_diff > 0`, `away_wins = 1` if `point_diff < 0`, `tie_flag = 1` if `point_diff == 0`
Compute Historical H2H:
For each game in `game_data`:
Filter `pbp_historical` for prior games between `home_team` and `away_team`
Count wins for `home_team`, `away_team`, and ties (2014–2023 + 2024 prior weeks)
Assign to `home_team_wins_10season`, `away_team_wins_10season`, `ties_10season`
Compute Team Performance Metrics:
For each team in each game:
Filter last 3 prior games from `pbp_historical`
Calculate wins, losses, avg points scored, avg points allowed
Filter all prior 2024 games for season record
Calculate wins, losses, and ties to date
Assign to `win_streak_last_3`, `losing_streak_last_3`, `wins_to_date`, `losses_to_date`, `ties_to_date`, etc.
Add Contextual Data:
Map teams to conferences and divisions
Set `same_conference`, `same_division` flags
Identify special games (e.g., Thanksgiving) from `schedule_2024`
Merge `gameday`, `gametime`, `spread_line`, `total_line`
Set `outdoor_game` based on `roof`
Post-Process:
Fill NaN values with 0
Round all numeric columns to 3 decimal places
Save to `NFL_DataScrap(2024-2025).csv`

4.2 Functioning Details

The script operates as follows:

- **Initialization**: Ensures all dependencies are installed and imported, printing version checks for debugging.
- **Data Ingestion**: Uses `nfl_data_py` to fetch raw play-by-play data (47,274 plays, 372 columns) and schedules, filtering for regular season to focus on Weeks 1–18 (272 games).
- **Preprocessing**: Adds boolean columns to the play-by-play DataFrame to identify possession and game periods, enabling conditional aggregations (e.g., yards when `home_poss` is True).

- **Aggregation**: Groups plays by game, applying functions like sum (e.g., `off_yards_gained_h`), mean (e.g., `def_epa_per_play_h`), and count (e.g., `total_plays`). Lambda functions filter data dynamically within groups.
- **H2H Logic**: Iterates over each game, querying historical data for prior matchups, counting wins and ties separately. Time complexity is O(n · m) where n is the number of 2024 games (272) and m is the number of historical games (2,500 over 10 seasons).
- **Performance Metrics**: For each team, retrieves the last 3 games for streaks and all prior 2024 games for season records, computing streaks, averages, and cumulative wins/losses/ties. Handles edge cases (e.g., Week 1 has no 2024 priors) by defaulting to 0.
- **Contextual Enrichment**: Merges schedule data and applies logical checks (e.g., `roof` mapping to `outdoor_game`), ensuring all games have complete metadata.
- **Finalization**: NaN handling ensures no missing values disrupt analysis, and rounding standardizes numeric precision to 3 decimals (e.g., 14.000, 0.123).

The output is a comprehensive game-level dataset (272 rows, 60+ columns), validated by inspecting key matchups (e.g., LA vs. ARI) and summary statistics.

5 Variables and Their Meanings

The dataset contains the following variables, aggregated per game:

Variable	Description
`season`	NFL season year (2024).
`week`	Week number of the regular season (1–18).
`game_id`	Unique identifier for each game.
`home_team`	Abbreviation of the home team.
`away_team`	Abbreviation of the away team.
`point_diff`	Final score difference (home score minus away score).
`home_win`	Binary indicator (1 if home team won, 0 otherwise).
`away_wins`	Binary indicator (1 if away team won, 0 otherwise).
`tie_flag`	Binary indicator (1 if game ended in a tie, 0 otherwise).
`off_yards_gained_h`	Total offensive yards gained by the home team.
`off_yards_gained_a`	Total offensive yards gained by the away team.
`off_pass_yards_h`	Offensive passing yards gained by the home team.
`off_pass_yards_a`	Offensive passing yards gained by the away team.
`off_rush_yards_h`	Offensive rushing yards gained by the home team.
`off_rush_yards_a`	Offensive rushing yards gained by the away team.
`off_turnovers_h`	Total offensive turnovers (fumbles lost + interceptions) by the home team.
`off_turnovers_a`	Total offensive turnovers by the away team.
`off_touchdowns_h`	Total offensive touchdowns scored by the home team.
`off_touchdowns_a`	Total offensive touchdowns scored by the away team.
`off_epa_per_play_h`	Average Expected Points Added per play for the home team offense.
`off_epa_per_play_a`	Average EPA per play for the away team offense.
`def_epa_per_play_h`	Average EPA per play for the home team defense (negative is better).
`def_epa_per_play_a`	Average EPA per play for the away team defense.
`off_wpa_h`	Total Win Probability Added by the home team offense.
`off_wpa_a`	Total WPA by the away team offense.
`def_wpa_h`	Total WPA by the home team defense (negative indicates preventing opponent wins).
`def_wpa_a`	Total WPA by the away team defense.
`off_cpoe_h`	Average Completion Percentage Over Expected for the home team offense.

off_cpoe_a	Average CPOE for the away team offense.
off_success_rate_h	Proportion of successful plays (EPA > 0) for the home team offense.
off_success_rate_a	Proportion of successful plays for the away team offense.
off_qb_epa_h	Total QB Expected Points Added for the home team offense.
off_qb_epa_a	Total QB EPA for the away team offense.
def_sacks_h	Total sacks recorded by the home team defense.
def_sacks_a	Total sacks by the away team defense.
def_qb_hits_h	Total QB hits by the home team defense.
def_qb_hits_a	Total QB hits by the away team defense.
off_red_zone_td_rate_h	Offensive touchdown rate in the red zone (yardline ≤ 20) for the home team.
off_red_zone_td_rate_a	Offensive red zone TD rate for the away team.
off_third_down_conv_rate_h	Offensive third-down conversion rate for the home team.
off_third_down_conv_rate_a	Offensive third-down conversion rate for the away team.
off_fourth_down_conv_rate_h	Offensive fourth-down conversion rate for the home team.
off_fourth_down_conv_rate_a	Offensive fourth-down conversion rate for the away team.
off_first_downs_h	Total offensive first downs achieved by the home team.
off_first_downs_a	Total offensive first downs by the away team.
off_pass_rate_h	Proportion of offensive plays that were passes for the home team.
off_pass_rate_a	Proportion of offensive passing plays for the away team.
off_shotgun_rate_h	Proportion of offensive plays run from shotgun formation for the home team.
off_shotgun_rate_a	Offensive shotgun play proportion for the away team.
off_no_huddle_rate_h	Proportion of offensive no-huddle plays for the home team.
off_no_huddle_rate_a	Offensive no-huddle play proportion for the away team.
timeouts_remaining_h	Timeouts remaining for the home team at game end.
timeouts_remaining_a	Timeouts remaining for the away team at game end.
score_differential_pre	Average score differential per play during the game.
total_plays	Total number of plays in the game.
first_half_plays	Number of plays in the first half (quarters 1 and 2).
second_half_plays	Number of plays in the second half (quarters 3 and 4).
overtime_plays	Number of plays in overtime (quarter 5+), typically 0 in regular season.
roof	Stadium roof type (e.g., "outdoors", "dome", "open", "closed").
surface	Playing surface type (e.g., "grass", "turf").
home_team_wins_10season	Home team's wins against the away team over 10 seasons (2014–2023 + 2024 prior weeks).
away_team_wins_10season	Away team's wins against the home team over 10 seasons.
ties_10season	Number of ties between the home and away teams over 10 seasons (2014–2023 + 2024 prior weeks).
home_win_streak_last_3	Number of wins by the home team in their last 3 games prior to this one.
away_win_streak_last_3	Number of wins by the away team in their last 3 games.
home_losing_streak_last_3	Number of losses by the home team in their last 3 games.
away_losing_streak_last_3	Number of losses by the away team in their last 3 games.
home_points_scored_last_3	Average points scored by the home team in their last 3 games.
away_points_scored_last_3	Average points scored by the away team in their last 3 games.
home_points_allowed_last_3	Average points allowed by the home team in their last 3 games.
away_points_allowed_last_3	Average points allowed by the away team in their last 3 games.
home_wins_to_date	Number of wins by the home team in the 2024 season prior to this game.
home_losses_to_date	Number of losses by the home team in the 2024 season prior to this game.
home_ties_to_date	Number of ties by the home team in the 2024 season prior to this game.
away_wins_to_date	Number of wins by the away team in the 2024 season prior to this game.

away_losses_to_date	Number of losses by the away team in the 2024 season prior to this game.
away_ties_to_date	Number of ties by the away team in the 2024 season prior to this game.
same_conference	Boolean: True if both teams are in the same conference (AFC/NFC).
same_division	Boolean: True if both teams are in the same division.
special_game	Boolean: True if the game is a designated special game (e.g., Thanksgiving, Black Friday).
gameday	Date of the game (e.g., "2024-09-15").
gametime	Kickoff time in Eastern Time (e.g., "20:20").
spread_line	Betting point spread (positive = home favored, negative = away favored).
total_line	Betting over/under total points line.
outdoor_game	Boolean: True if the game is outdoors (roof = "outdoors" or "open").

6 Legend for Team Abbreviations

The dataset uses the following standard NFL team abbreviations:

Abbr.	Team Name	Abbr.	Team Name
ARI	Arizona Cardinals	MIA	Miami Dolphins
ATL	Atlanta Falcons	MIN	Minnesota Vikings
BAL	Baltimore Ravens	NE	New England Patriots
BUF	Buffalo Bills	NO	New Orleans Saints
CAR	Carolina Panthers	NYG	New York Giants
CHI	Chicago Bears	NYJ	New York Jets
CIN	Cincinnati Bengals	PHI	Philadelphia Eagles
CLE	Cleveland Browns	PIT	Pittsburgh Steelers
DAL	Dallas Cowboys	SEA	Seattle Seahawks
DEN	Denver Broncos	SF	San Francisco 49ers
DET	Detroit Lions	TB	Tampa Bay Buccaneers
GB	Green Bay Packers	TEN	Tennessee Titans
HOU	Houston Texans	WAS	Washington Commanders
IND	Indianapolis Colts	LA	Los Angeles Rams
JAX	Jacksonville Jaguars	LAC	Los Angeles Chargers
KC	Kansas City Chiefs	LV	Las Vegas Raiders

7 Notes

- All float numeric values are rounded to 3 decimal places.
- Historical data (2014–2024) adjusts "STL" (St. Louis Rams) to "LA" for consistency.
- Overtime plays and ties are rare in the regular season dataset (`season_type == 'REG'`); ties are explicitly counted as `tie_flag`, `ties_to_date`, and `ties_10season`, excluded from wins and losses.
- Variable names follow a convention: offensive metrics are prefixed with `off_`, defensive with `def_`, followed by `_h` for home or `_a` for away (e.g., `off_wpa_h`, `def_sacks_a`).
- The script was developed and tested as of March 16, 2025, with continuously updated data from `nfl_data_py`.