**Synthetic Data Augmentation Documentation**

This document explains the techniques and methods used in the synthetic data augmentation pipeline. The goal of the augmentation is to generate realistic synthetic game data by combining multiple approaches: round robin scheduling, Monte Carlo simulation based on historical distributions, data perturbation, scenario-based adjustments, and a hybrid approach that integrates these methods.

**1. Overview**

The augmentation process creates synthetic match records by:

* **Pairing Teams:** Using a round robin approach to ensure every unique pair of teams faces off.
* **Simulating Performance Stats:** Using Monte Carlo simulation to generate game performance statistics based on each team’s historical data.
* **Perturbing Data:** Adjusting the generated statistics to account for factors like fatigue and injuries.
* **Scenario Adjustments:** Incorporating realistic game scenarios (e.g., home advantage, rest conditions, travel fatigue).
* **Hybrid Approach:** Combining these methods to form a robust simulation that mimics real-world variability.

**2. Augmentation Methods**

**2.1 Round Robin Method**

* **Purpose:** To ensure every team plays against every other team at least once.
* **Implementation:**
  + The code uses itertools.combinations(teams, 2) to generate unique team pairings.
  + For each pair, a synthetic match (or multiple rematches) is simulated.
* **Benefit:** Provides balanced coverage of matchups, important for statistical comparisons and scenario testing.

**2.2 Monte Carlo Simulation Based on Historical Distribution**

* **Purpose:** To generate performance statistics (e.g., field goals attempted, assists, turnovers) that follow the historical patterns of each team.
* **Implementation:**
  + For each team, the historical mean and standard deviation of key performance metrics are calculated.
  + The function simulate\_mc\_stats(team) uses these statistics to sample new values from a normal distribution.
  + When the standard deviation is zero (or not available), the mean value is used directly.
* **Benefit:** Captures the natural variation in team performance, ensuring that synthetic data reflects realistic variability.

**2.3 Data Perturbation**

* **Purpose:** To introduce controlled randomness and simulate external factors that affect game performance.
* **Techniques Used:**
  + **Fatigue Adjustment:**
    - Applies a 10% reduction in simulated statistics if the team’s rest days are below a threshold and travel distance exceeds a specified limit.
  + **Injury Adjustment:**
    - With a predefined probability (injury\_prob), simulated statistics are reduced by a factor (injury\_factor).
* **Benefit:** Mimics real-life perturbations such as player fatigue or injuries, making the synthetic dataset more robust and realistic.

**2.4 Scenario-Based Adjustments**

* **Purpose:** To incorporate situational factors that influence game outcomes.
* **Parameters and Adjustments:**
  + **Home Advantage:**
    - A multiplier (home\_advantage\_factor) is applied to the home team’s score.
  + **Game Fatigue and Travel:**
    - The simulation considers rest days and travel distance to adjust performance statistics.
  + **Overtime Simulation:**
    - Randomly simulates overtime minutes, representing additional game time scenarios.
* **Benefit:** Allows exploration of different “what-if” scenarios, such as the impact of playing on the road or with reduced rest.

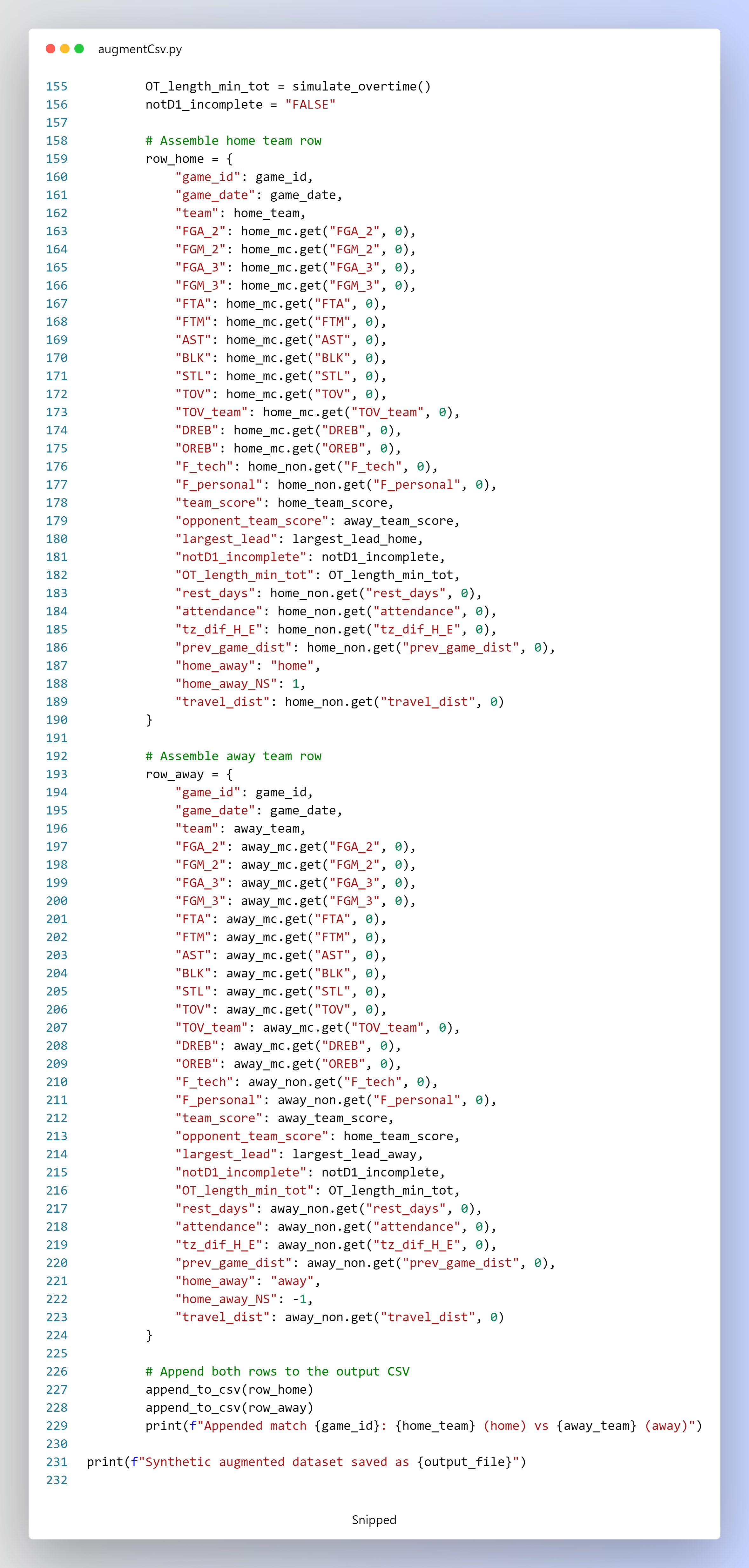
**2.5 Generative Models**

* **Purpose:** To generate synthetic data that follows a learned or observed distribution.
* **In This Context:**
  + While the code does not use a deep learning generative model (like GANs), it implements a generative approach by sampling from historical distributions using Monte Carlo methods.
* **Benefit:** Produces data that, while synthetic, remains statistically similar to real-world observations.

**2.6 Hybrid Approach**

* **Purpose:** To combine multiple augmentation techniques to leverage the strengths of each.
* **Implementation in the Code:**
  + The pipeline integrates:
    - **Round Robin Scheduling** (ensuring balanced matchups),
    - **Monte Carlo Sampling** (for generating performance stats),
    - **Data Perturbation** (for fatigue and injury scenarios),
    - **Scenario-Based Adjustments** (for home advantage and game conditions).
* **Benefit:** By combining these methods, the synthetic data better captures the complexity of real-world game scenarios, making it suitable for robust simulation and analysis.

**CODE OVERVIEW:**

**3. Code Walkthrough**

**3.1 Data Loading and Preprocessing**

* **Data Import:**  
  The CSV file (games\_2022.csv) is loaded, and missing numerical values are filled using column means.
* **Column Classification:**
  + **MC Columns:** Include key performance metrics used in the Monte Carlo simulation.
  + **Non-MC Columns:** Include additional game context data like rest days, attendance, and travel distance.

**3.2 Team Statistics Computation**

* **Historical Statistics:**  
  For each team, the historical mean and standard deviation for the Monte Carlo columns are computed.
* **Non-MC Averages:**  
  Historical averages for contextual columns are also stored for later use in simulation adjustments.

**3.3 Simulation Functions**

* **Monte Carlo Simulation (simulate\_mc\_stats):**  
  Generates simulated stats using a normal distribution based on historical means and standard deviations.
* **Non-MC Stats Retrieval (get\_non\_mc\_stats):**  
  Retrieves and rounds the historical averages for non-MC columns.
* **Score Calculation:**  
  Computes the team’s score using weighted contributions from field goals and free throws.
* **Overtime Simulation:**  
  Simulates overtime minutes with a high probability of zero overtime.

**3.4 Match Generation**

* **Team Pairing:**  
  Uses round robin (via itertools.combinations) to select team pairs.
* **Home/Away Assignment:**  
  Randomly assigns the home and away status, applying home advantage to the home team’s score.
* **Scenario Adjustments:**  
  Applies fatigue and injury adjustments based on predefined thresholds and probabilities.
* **Output Assembly:**  
  Two rows (one per team) are generated for each synthetic match and appended to the output CSV file.

**3.5 Output**

* **Real-Time CSV Appending:**  
  Each synthetic match row is appended to the output CSV file (aug\_round\_robin\_montecarlo\_scenario.csv), allowing incremental data generation and storage.

**4. Conclusion**

This synthetic data augmentation pipeline leverages a **hybrid approach** to create realistic game data by integrating:

* A **round robin method** for balanced matchups,
* **Monte Carlo simulations** for performance variability,
* **Data perturbation** to simulate fatigue and injuries,
* **Scenario-based adjustments** to account for game-specific factors.

The result is a robust and flexible synthetic dataset that mirrors the nuances of real-world game situations and can be used for further analysis, modeling, or simulation studies.