**Model summary**

**Issues:**

The current statistics for this game (Kill Team) are very poor – to the extent there is no valid robust ranking of either players or teams.

**Objective:**

To use a Machine Learning model to determine:

* The relative strength (i.e. a ranking) of the different/distinct armies (as per armyName/ armyId)
* Provide a ranking of player skill, indepent of army used

Both methods must produce tested and valided predictions

**Output**

1. Army strength ranking
2. Player skill ranking
3. To enable me to reliably predict the outcome (i.e. the Position 1st, 2nd, 3rd to the last place) of a given tournament knowing the players who entered and their respective teams

Ensuring all outputs are tested, validated, reliable, replicatable if new .json data set is used

I need you to propose a Machine Learning model - initially just theory, logic, structure (no code) of how one might do this

**Model Structure:**

To tackle this challenge, we need to break down the problem into three interconnected parts: ranking armies, ranking player skill independently of the army, and predicting tournament outcomes. Here’s a proposed approach to the Machine Learning (ML) model structure:

**1. Relative Army Strength Ranking**

* **Objective**: Determine the inherent strength of each army (armyName/armyId) independently of the players using them.
* **Input Features**:
  + Army ID (armyId)
  + Tournament-specific attributes (e.g., number of rounds, total players)
  + Performance metrics in games (e.g., average points scored per game, win/loss ratio)
* **Target Variable**: Army effectiveness score (derived from a performance metric such as win rate, average placing, or points accumulated per tournament).
* **Model**:
  + **Model Type**: A regression model like a Gradient Boosting Regressor or a neural network that maps army characteristics to an effectiveness score.
  + **Approach**:
    - Aggregate performance metrics for each army across all tournaments.
    - Use tournament attributes (total players, rounds) and army attributes as features.
    - The model will learn to associate these attributes with an army's relative performance.
* **Validation**:
  + Split data into training and validation sets based on tournaments.
  + Use cross-validation to ensure that the model is generalizing well to different tournaments.
  + Compare predicted effectiveness scores against actual performance (win rates, average placing).

**2. Player Skill Ranking (Independent of Army)**

* **Objective**: Assess player skill without the influence of the army they are using.
* **Input Features**:
  + Player ID (userId)
  + Historical performance in tournaments (e.g., total points, game results).
  + Opponent information (i.e., quality of opponents faced).
  + Army used in each game.
* **Target Variable**: Player skill score or rank (can be derived from player performance metrics).
* **Model**:
  + **Model Type**: A collaborative filtering or matrix factorization approach, similar to recommender systems.
  + **Approach**:
    - Treat the player-tournament interaction as a "rating" problem. The player’s performance (win/loss, points) in each tournament becomes the "rating."
    - Use a factorization machine or a collaborative filtering model to extract latent factors that represent player skill and army strength.
    - By including army information as an input, the model can separate the player’s inherent skill from the army's strength.
* **Validation**:
  + Cross-validation on player performance data across multiple tournaments.
  + Check consistency of predicted skill scores over time.

**3. Predicting Tournament Outcomes**

* **Objective**: Predict the ranking of players in a tournament based on their skill and army strength.
* **Input Features**:
  + Player skill scores (from the player ranking model).
  + Army strength scores (from the army ranking model).
  + Tournament attributes (total players, rounds, etc.).
* **Target Variable**: Predicted tournament rankings for each player (ordinal position: 1st, 2nd, etc.).
* **Model**:
  + **Model Type**: An ordinal regression model (e.g., Ordinal Logistic Regression) or a probabilistic ranking model (e.g., Plackett-Luce model).
  + **Approach**:
    - Combine player skill and army strength as inputs.
    - Use tournament attributes to adjust predictions.
    - The model predicts the likelihood of each possible rank for each player in a tournament.
* **Validation**:
  + Use historical tournament data to compare predicted rankings with actual outcomes.
  + Evaluate performance using metrics like rank correlation (Spearman’s rank) to measure how well the predicted ranks match the actual tournament results.

**Overall Strategy**

1. **Feature Engineering**:
   * Aggregate performance metrics for each army and player.
   * Include contextual tournament data as input features (e.g., total players, rounds).
   * Normalize performance metrics to account for differences in tournament structure.
2. **Modular Training**:
   * Train separate models for army strength and player skill first.
   * Use outputs from these models as input features for the tournament outcome prediction model.
3. **Validation and Testing**:
   * Use cross-validation at each model training step to ensure robust performance.
   * Simulate tournament scenarios to test the predictive accuracy of the final tournament outcome model.
4. **Generalizability**:
   * Design the system to ingest new JSON datasets, extract the relevant features, and update the models as more data becomes available.
   * Employ techniques like incremental learning if the models need to be updated frequently with new data.

This approach provides a layered analysis:

* **Army strength** is quantified independently of player skill.
* **Player skill** is evaluated independently of the army they use.
* **Tournament outcome predictions** combine both player and army assessments to rank potential outcomes, ensuring the model is robust, validated, and adaptable to new datasets.