***PLAYSIMPLE GAMES ASSIGNMENT SUBMISSION***

By, Hemanth Kumar NVS

USN: 1MS22AD027

**Chosen problem statement: Q2) Matchmaking for the users**

**Elaboration:** There are some games in which players play against each other. These are called Player vs Player (PvP) games. Assume you are a newly hired Data Scientist assigned to build matchmaking algorithm for such a game. How would you go about making a matchmaking algorithm? How will you define this was a good match or a bad match? What should be player-centric measures or business metrics that will help decide matchmaking algorithm is performing good/bad. Assume that as of today, the game team had implemented random matchmaking algorithm allowing you to gather data.

**Proposed solution:**

**1. Research on the problem:**

1. Relevant blog post by some game developers (few of them):
   * <https://www.gameanalytics.com/blog/matchmaking-tips-for-game-developers>
   * <https://joostdevblog.blogspot.com/2014/11/why-good-matchmaking-requires-enormous.html>
   * <https://gametree.me/gaming-terms/matchmaking/>
2. Briefly summarizing the relevant research papers using AI tools:
   * <https://www.cemyuksel.com/research/matchmaking/i3d2024-matchmaking.pdf>
   * <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3928966>
3. Video explanations on various platforms, like YouTube for better understanding.
4. Real time collection of data and opinion from multiple players.

1. **Game selection for solving:**

The game considered for solving this problem is Chess!

Reason: It is one of the classic games and Chess is ideal for matchmaking research as it’s purely skill based, **uses established rating systems**, offers structured data, and has diverse time formats and a global player base and it is one of the games which cannot be played without an opponent. Hence, chess is my choice.

1. **Assumptions and approach:**

Assumptions:

* + Every user skill is **normalized** to a common system for measuring the individual skill which is the **Elo rating**.
  + Initially, players make a clear **choice** between **ranked** and **unranked** matchmaking modes.
  + **Geographic proximity** is important in matchmaking because latency detracts from the gameplay experience, particularly in fast-paced time controls.
  + It is possible to learn player behaviour patterns from past gameplay data and encode them into a **behavioural profile**, such as considering their history of the last 10 games.

Approach:

* + **Skill Normalization and Player Choice:**

Normalize player skill using a common rating system (e.g., Elo rating). Allow players to clearly choose between ranked and unranked matchmaking modes to set their gameplay intent.

* + **Performance profiling – Considering the last 10 games:**

Using the last 10 games captures recent performance trends, helping to adjust matchmaking dynamically, avoid unfair pairings, and reflect a player's true current skill level.

* + **Latency-Aware Matching:**

Factor in geographical proximity and latency to ensure smooth gameplay, especially in fast pacedformats where delays significantly affect performance.

* + **Ranked matchmaking:**

In ranked mode, ensure skill-based and fair matches in the random matchmaking mode that impact a player’s Elo rating. Offer two choices:

Private Matches: Invite a friend for a non-impactful, personal game.

Random Matchmaking: Match players using KNN-based similarity across multiple dimensions.

1. **Datasets needed to solve the problem:**

To address this problem, we employ the K-Nearest Neighbours (KNN) algorithm by mapping each player into a three-dimensional space. The following parameters are essential for constructing the dataset required for this solution:

* + **Current Elo rating -** Represents an average Elo of that last 10 "game specific Elo’s" of the player
  + **Win Rate:** Proportion of games won out of the player's last 10 matches.
  + **Geolocation** - Approximate player location (e.g., country or region) used to reduce latency and improve game performance.

These features enable the model to efficiently compute proximity in the **3D space** and identify the most suitable opponents within a defined similarity.

Few datasets that can be taken into consideration and relevant are:

* + Custom tailored dataset for this approach <https://tinyurl.com/ymkxb34m> (Made using ChatGPT)
  + <https://www.kaggle.com/datasets/arevel/chess>[-games](https://www.kaggle.com/datasets/arevel/chess-games)
  + <https://www.kaggle.com/datasets/datasnaek/chess>
  + Another great source which is the <https://stockfishchess.org/>datasets, etc.

If additional information is required for further development and is not readily available in the existing dataset, it can be efficiently gathered through **brief in-game surveys**. These surveys can be designed to collect relevant player preferences or behavioural insights, ensuring minimal disruption to the user experience while enhancing the accuracy and effectiveness of the matchmaking system.

1. **Algorithms considered to solve the problem and its implementation:**

Essentially, our algorithm is designed to operate when a player selects the random match option. The proposed solution aims to optimize matchmaking by leveraging the **KNN approach**, ensuring that players are paired with the most compatible opponents based on key parameters.

In the Random Matchmaking the KNN-Based Matchmaking: Every player in the matchmaking pool is shown as a point in three dimensions, with the following axes:

* + Rating of Elo Skill based on the average of the past 10 games.
  + Win rate based on the past 10 games.
  + Geographical Position (with respect to the nearest server)

Here’s a **step-by-step explanation** of how the algorithm solves the matchmaking problem:

* 1. **Player Data Collection:**

When a player opts for a random match, their relevant data—such as Elo rating, gameplay aggressiveness, and geolocation—is collected.

* 1. **Feature Mapping:**

Each player is represented as a point in a three-dimensional feature space, where each axis corresponds to one of the key parameters.

* 1. **Distance Calculation:**

The algorithm computes the distance between the active player’s point and all other players currently available for matchmaking, typically using a suitable distance metric (e.g., Euclidean distance).

* 1. **Neighbour Selection:**

Using the K-Nearest Neighbours method, the algorithm identifies the closest players within a predefined threshold distance to ensure similarity in skill, playstyle, and location.

* 1. **Match Confirmation**:

From the identified neighbours, the best match is selected based on additional criteria if needed (e.g., latency, recent activity), and the match is confirmed.

* 1. **Fallback Handling:**

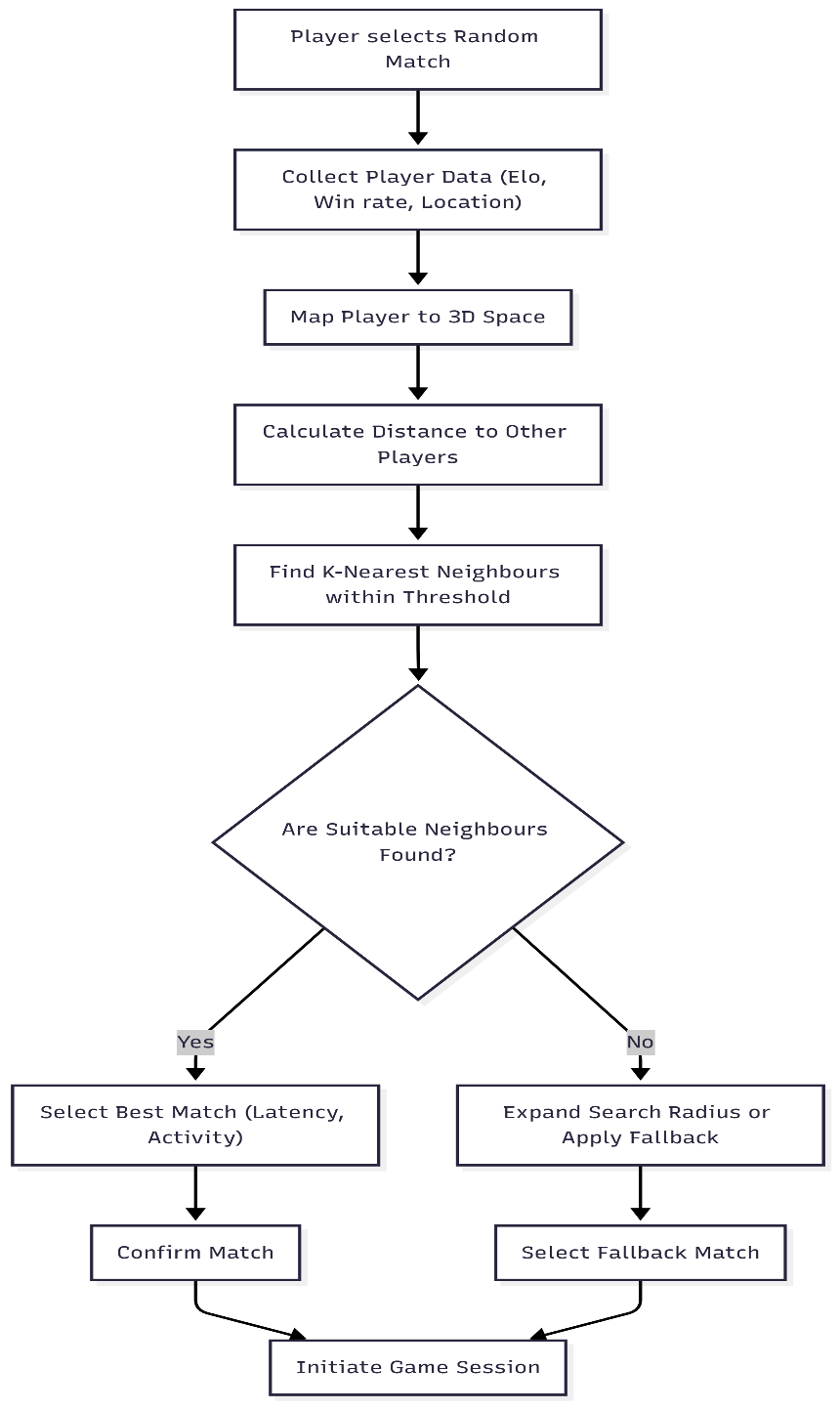
If no suitable neighbour is found within the threshold, the algorithm can expand the search radius or default to a broader matchmaking rule to minimize player wait times.

* 1. **Match Initiation:**

Once the best match is identified and confirmed, the game session is initiated, providing the player with a balanced and optimized opponent.

This process ensures fair, efficient, and regionally optimized matchmaking, improving player experience during random match selections.

**Flowchart:**



1. **Proving the proposed solution is working positively and not impacting in the negative way:**

To ensure that the new matchmaking algorithm delivers a positive impact without introducing negative effects, a structured validation approach is used.

* + **Classic A/B Testing:**

We compare the new algorithm (variant group) against the existing system (control group) in a live environment. Metrics such as the difference in the average Elo rating of the opponents, difference in the accuracy of the players in the game played, retention, session duration, number of moves, game outcome and churn are tracked. A statistically significant improvement in these metrics confirms positive impact.

* + **No Negative Trade-offs**:

Alongside improvements, we ensure there are no regressions in critical areas like:

* Increased wait time in queue
* Drop in monetization
* Rise in early match exits
  + **Balanced Evaluation:**

Both player-centric metrics (e.g., satisfaction, retention) and business metrics (e.g., monetization, churn) are assessed to ensure holistic improvement.

* + **Continuous Monitoring**:

Post deployment dashboards and feedback loops are established to detect anomalies or declines early, allowing timely rollback or refinement.

* + **Taking surveys** and opinions from the players, because ultimately, they are the **decision makers**.

This approach ensures the solution is both effective and sustainable, with clear evidence backing its performance.