Comparative Fantasy Basketball Player Algorithm

Overview

We are given two NBA players, each described by the following metrics returned by the model:

- P: player name
- S: injury/availability status
- **ProjList**: list of projected fantasy scores for the next 5 games
- ProjAvg: average fantasy projection over next 5 games
- Trend: qualitative recent momentum
- Risk: risk factor (low = safer)
- **Last5**: statistics from their last 5 real NBA games
- PMAvg: average plus/minus over the last 5 games
- Pos: player position
- ID: NBA player identifier

The goal is to construct a *comparative algorithm* that determines which of two players A and B is a better fantasy pick.

Our guiding principles:

1. Fantasy projection is the most important predictor.
2. Recent performance and momentum must modify projections.
3. Risk reduces trust in projection.
4. Player availability is a hard constraint (injured $\Rightarrow$ auto-lower score).

In my MATH135, we learned how to derive equations using Sets. I will use the guiding principles to do so:

1. Interpreting the Metrics

Projection Average

Let

$$\text{ProjScore}_i = \text{ProjAvg}_i$$

for player $i \in \{A, B\}$. This captures the expected raw output.

Trend Adjustment

Define a numerical mapping:

$$\text{TrendBoost}_i = \begin{cases} +5 & \text{if trend is "hot" or "going crazy"}, \\ +2 & \text{if trend is "positive" or "improving"}, \\ 0 & \text{if neutral}, \\ -3 & \text{if cold}. \end{cases}$$

Risk Penalty

In my program a Risk score of 100 is the safest. High risk is better. Define:

$$\text{RiskPenalty}_i = 1/2 \cdot \text{Risk}_i.$$

Plus/Minus Context

High plus/minus reflects quality of role and team environment. Define:

$$\text{PMScore}_i = 0.4 \cdot \text{PMAvg}_i.$$

Status Adjustment

Let status S_i be:

$$\text{StatusPenalty}_i = \begin{cases} +0 & \text{if active}, \\ -100 & \text{if out}. \end{cases}$$

2. Final Fantasy Fitness Score

We combine all weighted effects into a single value:

$$\text{FFS}_i = \text{ProjScore}_i + \text{TrendBoost}_i + \text{PMScore}_i + \text{StatusPenalty}_i - \text{RiskPenalty}_i.$$

The comparison rule is:

$$\text{Pick } \begin{cases} A & \text{if FFS}_A > \text{FFS}_B, \\ B & \text{otherwise}. \end{cases}$$

3. Justification of the Model

Projection Dominance

Projections encapsulate both the player's historical production and matchup context. Thus they form the primary term.

Momentum (Trend) Importance

Trends influence coaching decisions, usage rate, and player confidence. A hot streak often persists for several games.

Risk Dampening

Risk factor represents volatility. A high-risk player may have explosive games but reduces reliability.

Plus/Minus as Stability Signal

Players with strong plus/minus averages are less likely to be benched and more likely to receive stable minutes.

Status as a Hard Constraint

A player who is out cannot help. Questionable players reduce trust.

4. Pseudocode

```
function FantasyFitnessScore(player):
proj = player.projection_avg
trend = mapTrendToBoost(player.trend)
risk_penalty = 2 * player.risk_factor
pm = 0.4 * player.last5_plus_minus_avg
status_penalty = mapStatus(player.status)

return proj + trend + pm + status_penalty - risk_penalty


function ComparePlayers(A, B):
scoreA = FantasyFitnessScore(A)
scoreB = FantasyFitnessScore(B)

if scoreA > scoreB:
return A
else:
return B
```

Conclusion

The proposed comparative algorithm balances:

- raw projection,
- short-term trend,
- risk,
- lineup stability,
- availability status.

This creates a mathematically interpretable and computationally simple decision rule suitable for fantasy basketball applications.