**Football Player Recommendation System**

**Overview**

The goal of this project is to build a system that can recommend the best replacement football players for a given position on a team. The system will leverage machine learning and data science techniques to analyze player data and make optimized recommendations.

The target users are football club coaches who want to build a new team from scratch or replace a player on their existing team. The system will have two main interfaces:

1. **Build New Team Interface**: Allows the coach to specify which positions they need filled. The system will recommend the best available players for those positions based on analysis of all candidate players.
2. **Replace Player Interface**: Allows the coach to select a player on their current team they want to replace. The system will recommend the best replacement player from the pool of available candidates.

**Data**

The system will utilize the following data:

* **Player profiles**: Contains attributes for each player including position, nationality, number of international matches/goals, junior level experience, etc. This includes registered players and new players who have submitted a registration profile.
* **Team rosters**: Lists the current players on each team's roster. Used to filter out players already registered to other teams.
* **Rejected players**: List of players rejected by clubs. These players will be clustered with new player registrations as candidate replacement players.

**Algorithms**

The main algorithms will include:

* **Player clustering**: Unsupervised clustering models like K-Means will group players based on attributes like position, skills, experience, etc. This will allow querying clusters for replacement options.
* **Player recommendation**: Models like content-based filtering will analyze player profiles and team needs to recommend the best additions. Models will be optimized for metrics like team balance, player compatibility, and skill level.
* **Search/ranking**: Best replacement players will be efficiently searched and ranked based on relevance. This may leverage algorithms like KNN search.

**System Architecture**

The system will follow a standard web-based architecture:

* **Frontend**: HTML/CSS/JavaScript frontend for the web interface. Allows coaches to specify needs and view recommendations.
* **API backend**: Python/Flask backend API server. Handles model predictions and data processing.
* **Database**: MongoDB database to store player profiles, rosters, etc.
* **Machine learning**: Python model training and prediction scripts. Loads data, trains model, makes predictions.

The frontend will call API endpoints on the backend server to retrieve model predictions and power the interfaces. The backend will load data from the database and call ML models to generate recommendations.

**Evaluation**

The system will be evaluated on:

* **Prediction accuracy**: Ability to accurately recommend players that improve team composition and performance. Measured against expert picks and team outcomes.
* **User satisfaction**: Ease-of-use, quality of recommendations, and utility for coaches. Assessed via user studies and surveys.
* **Model optimization**: Precision and speed of search and retrieval. Models continuously improved via offline tuning and evaluation.

**Future Work**

Potential extensions for future development:

* Incorporate additional data like player salaries, team budgets, team chemistry etc.
* Expand models to other sports and scenarios like draft pick selections.
* Add data visualization tools to analyze players, teams, and recommendations.
* Move to production deployment on cloud infrastructure.