

From Ratings to Rising Stars: Predicting Player Potential Using FIFA Attributes

Jonas Schrade (01/1080887) & Niklas Bacher (01/1081503)

Introduction

This project uses the European Soccer Database with a specific focus on player-level information. A central component of the dataset is the set of player attributes derived from the FIFA video game series. These attributes summarize players' technical skills, physical characteristics, and overall ability as assessed by professional scouts and analysts. Although the ratings originate from a simulation game, they are widely used as standardized and comparable measures of player quality and have been shown to correlate strongly with real-world performance. As such, they represent an accessible and realistic information source for clubs, analysts, and researchers interested in player evaluation.

From an applied perspective, accurately predicting player performance based on such attributes has clear practical relevance. Clubs can use these predictions to support recruitment decisions, contract negotiations, or talent development, while analysts may rely on them to benchmark players across leagues and seasons. Understanding the predictive value of FIFA attributes therefore has direct implications for real-world decision-making in professional football.

The goal of this project is to assess whether complex models such as deep neural networks provide meaningful improvements over simpler statistical learning approaches in mapping player characteristics to observed performance measures and potential ratings. Player potential is inherently a latent concept: it reflects expected development and long-term ability rather than directly observed outcomes. Capturing position-specific differences that are not explicitly encoded in the input variables may favor more flexible models such as neural networks. To evaluate this, a traditional benchmark model is used as a baseline and compared against a neural network using the same predictors and outcome. This comparison allows us to determine whether deep learning provides added value in identifying players with high potential, or whether simpler models already capture most of the relevant signal contained in the FIFA-based player attributes.

Analysis

The analysis draws on player-level observations from the European Soccer Database, combining repeated measurements of FIFA-derived player attributes with assigned ratings of potential. The dataset contains a rich set of numeric variables capturing technical skills, physical abilities, and position-specific competencies, complemented by basic player characteristics such as age and preferred foot.

Data Preparation

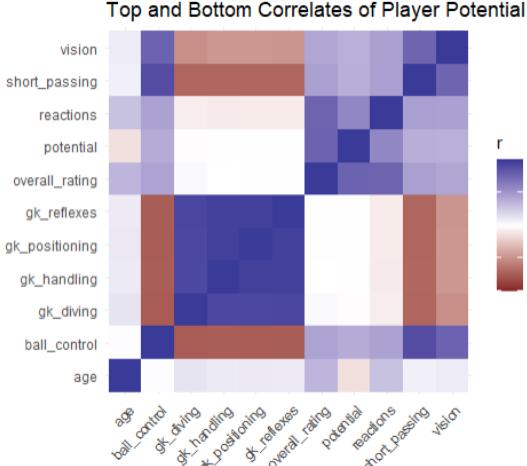
Several preprocessing steps were applied prior to analysis. Measurement dates and birthdates were used to compute each player's age as the number of full years at the time each attribute snapshot was recorded. Observations corresponding to players younger than 16 were excluded to remove implausible or non-comparable records, which were identified in a manual screening. Non-informative identifiers and metadata, including player names, internal IDs, and FIFA-specific API keys, were dropped to retain only variables relevant for prediction.

To ensure compatibility with the modeling framework, observations with missing values were removed. Overall missingness in the dataset is low, with only a small subset of technical attributes exhibiting minor gaps. Given the large sample size and the focus on predictive performance rather than inference, a complete-case approach was adopted. When modeling player potential, the overall rating was excluded from the feature set to avoid introducing mechanically related information into the prediction task. The final dataset contains 178,364 observations and 34 features from 10,582 players.

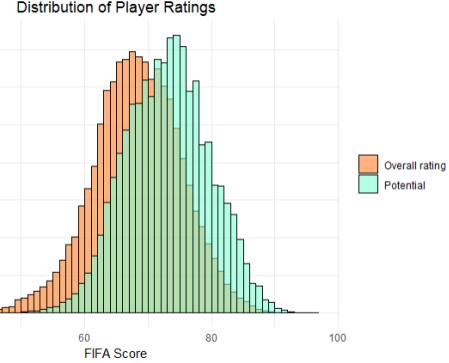
Variable Structure and Correlations

The resulting dataset consists primarily of continuous numeric predictors bounded between 0 and 100, reflecting FIFA's attribute scoring system, alongside a small number of discrete or binary variables such as preferred foot. Correlation analysis reveals that player potential is strongly associated with several technical and cognitive attributes, including vision, short passing, reactions, and ball control. These relationships align with the role of such skills in evaluations of long-term player quality.

In contrast, goalkeeper-specific attributes exhibit weak or negative correlations with potential when considering the full sample. This pattern reflects the strong positional structure of the data: attributes relevant for goalkeepers carry little information for outfield players and vice versa. The correlation heatmap highlights this divide and suggests that interactions between player position and skill attributes may be important for prediction, even when position itself is not explicitly modeled.



(a) Figure 1: Correlations among relevant predictors and potential score



(a) Figure 2: Distribution of overall and potential player ratings

Distributions of Attributes and Ratings

The marginal distributions of player attributes are generally unimodal and concentrated around mid-to-high values. Goalkeeper attributes display different distributions, with substantial mass at low values, reflecting the fact that most players in the dataset are not goalkeepers. The preferred foot variable is highly imbalanced, with right-footed players forming a clear majority.

The distributions of overall rating and potential, shown in Figure 2, further illustrate the distinction between current performance and assessed future ability. While the two measures are closely related, potential ratings are systematically shifted toward higher values, indicating that many players are evaluated as having greater long-term upside than their present level of performance suggests. This gap motivates the focus on potential as a separate prediction target and underscores the challenge of modeling a concept that is inherently forward-looking and position-dependent.

Methods

Neural Network

The neural network follows a sequential architecture consisting of an input layer matching the feature dimensionality, three fully connected hidden layers with ReLU activations, and a single-unit output layer with linear activation. Each hidden layer incorporates L2 (ridge)

regularization with a decay of 0.001, batch normalization, and dropout to constrain weights, accelerate convergence, and enhance robustness, respectively.

For model training, features are organized in a numeric design matrix to ensure consistent encoding of categorical variables, with the target variable stored separately. Training and evaluation employ nested, group-aware cross-validation, grouping folds by unique player identifiers to prevent cross-sectional data leakage. Within each training fold, hyperparameters are selected via grid search using an internal train-validation split. The grid varies hidden-layer sizes, dropout rates, and the learning-rate annealing factor (two values each; see Table ...), while batch size (128), maximum epochs (50), and early stopping with a patience of five epochs are held constant across all folds and configurations. The Adam optimizer with a default initial learning rate of 0.001 is used throughout, enabling fast and stable convergence. This approach balances robust model validation with computational efficiency and reproducibility. Before training, 20% of the training data are held out as a validation set to guide optimization and hyperparameter selection. The normalization scaler is fitted exclusively on the inner training set and subsequently applied to the validation and test sets.

Hyperparameter	O1	O2	F1	F2	F3	F4	F5
Size Layer 1	256	128	128	128	128	128	128
Dropout 1	0.4	0.3	0.4	0.3	0.3	0.4	0.3
Size Layer 2	128	64	64	128	128	128	128
Dropout 2	0.3	0.2	0.2	0.3	0.2	0.3	0.2
Size Layer 3	64	32	64	32	64	64	64
Dropout 3	0.2	0.1	0.1	0.1	0.1	0.1	0.1
LR Annealing	0.5	0.1	0.5	0.1	0.5	0.1	0.5

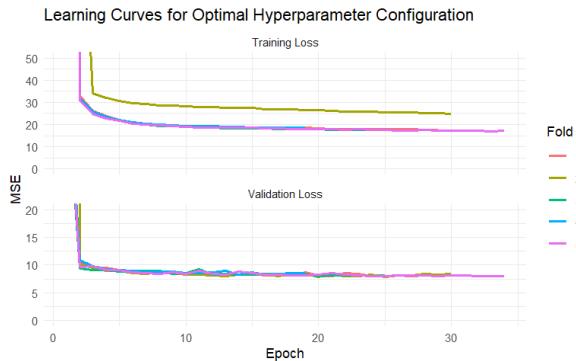


Figure 3: Learning curves of the optimal hyperparameter configuration across five outer folds.

Across five folds, 128 hyperparameter combinations are evaluated, resulting in 640 training runs. The learning curves of the optimal configurations exhibit a sharp initial decline in

both training and validation loss, followed by a gradual decrease and a clear plateau in the validation loss around epoch 30, which triggers early stopping. Notably, in fold 2 the training loss stabilizes more rapidly after the initial decline, suggesting greater structural complexity in the training data. Finally, the optimal model is tested against the test fold with evaluation metrics reported below.

Baseline model: Lasso Regression

As a transparent benchmark, we estimate a linear prediction model for player potential using Lasso regression. The Lasso extends OLS by adding an ℓ_1 penalty that shrinks coefficients toward zero and sets some exactly to zero, thereby regularizing the model and performing variable selection. As in the neural network setup, all predictors are converted into a numeric design matrix. Predictors are standardized during estimation so that the penalty treats all features comparably across scales.

Model performance is evaluated using grouped 10-fold cross-validation. As before, folds are constructed at the player level to prevent information leakage across training and test sets. We use 10 folds rather than LOOCV for computational efficiency, as training and test errors varied only marginally across folds, indicating stable performance estimates.

Within each training fold, the regularization parameter λ is selected via internal cross-validation over a grid of candidate values. The model is then refit using the selected λ , and predictive performance is assessed using MSE and MAE on training and held-out test data.

Finally, we conduct diagnostic checks for the linear baseline, including assessments of linearity, homoskedasticity, and the residual distribution. While these conditions are not strictly required for predictive performance, particularly in the case of normally distributed errors, the diagnostics indicate that they are reasonably well satisfied and provide useful context for comparing the linear model to more flexible approaches.

Results

Discussion of how your model performed. Include a discussion about whether or not Deep Learning was necessary in this situation. Note: I do not want you to include large chunks of R output, just summaries that explain your model and its performance sufficiently. One of the marking criteria is whether you can do this in a structured way.

If you want a table you can make one with [this website](#) and paste the markdown table here. For example:

Averaged Evaluation Metrics	Train MSE	Test MSE	Train MAE	Test MAE
Neural Network	7.36	7.91	1.94	2.01

Averaged Evaluation Metrics	Train MSE	Test MSE	Train MAE	Test MAE
Lasso Regression	15.83	15.90	3.07	3.08



Figure 4: My Caption Here

(Note that the `width=300` argument controls how wide your image will be.)

Reflection

Reflections on what you learned/discovered in the process of doing the assignment. Write about any struggles you had (and hopefully overcame) during the process. Things you would do differently in the future, ways you'll approach similar problems in the future, etc.