Football Players Pricing

Author: Blibeche Farès
Date: January 2026

This study explores the prediction of football players' market value using data extracted from the FIFA19 player database. Using a dataset of 17,954 players and more than 450 engineered features, three regression models were tested: linear regression, LASSO regression, and a neural network. Model performance was evaluated using RMSE on both the raw price scale and the logarithmic scale. After log-transformation of player values, the linear regression model achieved the lowest test RMSE (0.037), outperforming LASSO (0.039) and the neural network (0.086). This corresponds to an pricing error of 1€, approximately $4x10^{5}$% of averrage market value of players in dataset, showing that log-scale modeling provides reliable and stable predictions for automated football player pricing.

1. Introduction

Since the 2010s, football clubs like Brighton have increasingly relied on data to improve recruitment. In this data-driven era, players have become key financial assets traded between clubs, making accurate pricing essential. This project adopts the perspective of a club analyst aiming to build a model that can automatically assign player values from available attributes (release clauses, performance, age, etc.). Due to the lack of internal training data, the study uses the 2018/2019 FC game database (formerly FIFA) as a proxy for real performance metrics.

Goal : to price correctly football players using known informations about their release clauses, performance, physical features ...
Because player value in euro is a continuous variable, this is a regression.

2. Data

This dataset, obtained by scraping the website SoFIFA.com, contains 51 features about player performance, market value, club affiliation, playing position, and even each player's preferred foot, for each player recorded in the FIFA game database during the 2018/2019 season (N = 17,954, see **Figure 1**).

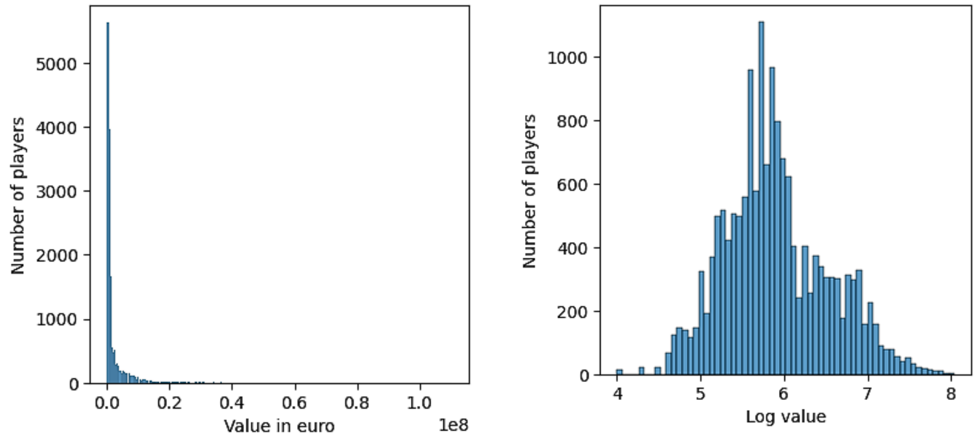


Figure 1 – Distribution of players value in € (a.) and logarithm of value (b.)

We can see that log value is more readable than value itself. We will compare our predition on both variables.

3. Method

I created more than 400 new variables by computing the square of every quantitative variable, and by turning all categorial variables into many binary variables. I then standardized the features and separated the data into training set (75%) and testing set (25%). I finally built and evaluated three regression models, using *scikit-learn* python3 library:

- Linear regression
- LASSO regression (5-fold cross-validated for 35 fits, 1 / λ ∈ [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1])
- Neural Network (5-fold cross-validated for 720 fits and several hyper-parameters, see **Table A1**)

I aim to compare models based on their ability to minimize large prediction errors, I then use the RMSE which penalizes larger deviations. I will process thoses 3 methods two times, one time to predict value, and a second time to check if log value is easier to predict.

4. Results

4.1. Results for value prediction

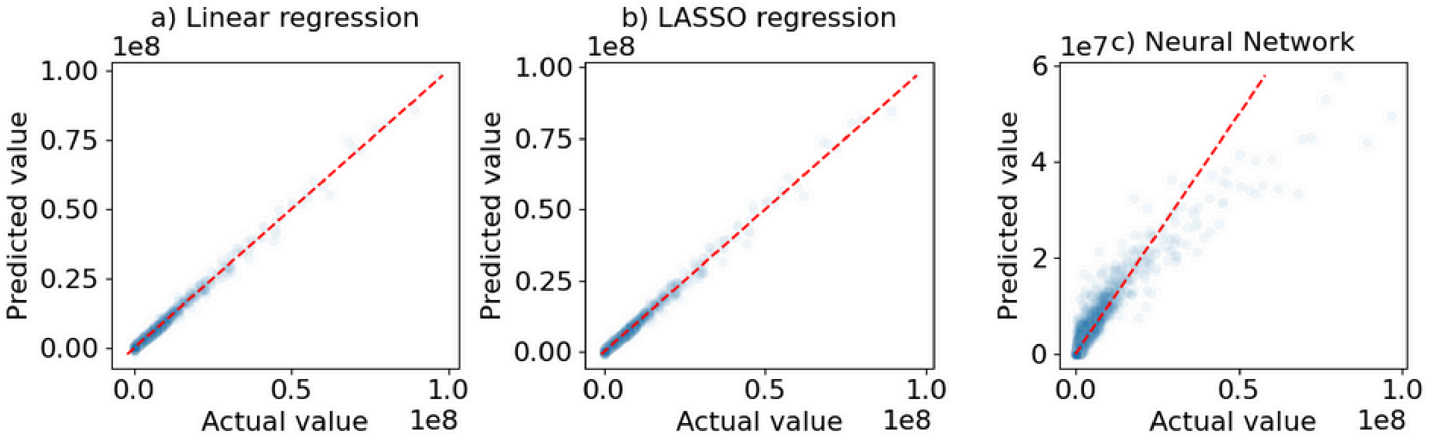


Figure 2 – Scatterplot of predicted versus actual values of age for (a) *Linear regression*, (b) *LASSO regression* and (c) *Neural Network*

Model	Train RMSE	Test RMSE
Linear regression	439,168	493,034
LASSO regression	467,712	527,198
Neural Network	2,107,726	2,403,101

Table 1 – Summary of train RMSE and test RMSE (for value)

4.1.1. Linear regression

The RMSE of this model is 439,168 € on the training set, and increase slightly to 439,034 € on the testing set (see **Figure 2.a** and **Table 1**). This small gap suggests low variance. The high value of test RMSE (17% of averrage value) indicates that the model may from bias.

4.1.2. LASSO regression

The RMSE of this model is 467,712 € on the training set, and increase to 527,198 € on the testing set (optimized 1 / λ = 0.001, see **Figure 2.b** and **Table 1**). This small gap suggests low variance. The high value of test RMSE (21% of averrage value) indicates that the model suffer from bias.

4.1.3. Neural Network

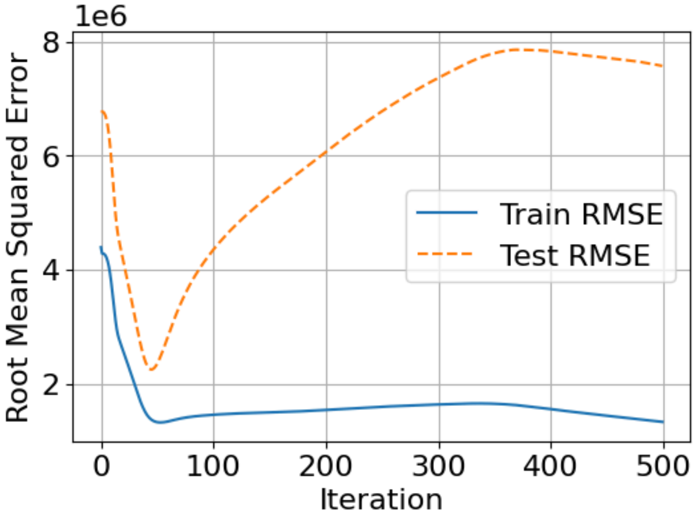


Figure 3 – Error curve of the neural network

The RMSE of this model is 2,107,726 € on the training set, and increase to 2,403,101€ on the testing set (see **Figure 2.c** and **Table 1**, see also **Table A1** for optimized hyperparameters). This hudge gap suggests very high variance. The excessively high value of test and train RMSE (more than 85% of averrage value) indicates that the model suffer from a lot of bias. The error curve show an optimum around 50 iterations (**Figure 3**), indicating that the model minimize error with an early stopping. But test RMSE stay high, it's maybe due to overfitting.

4.2. Results for log value prediction

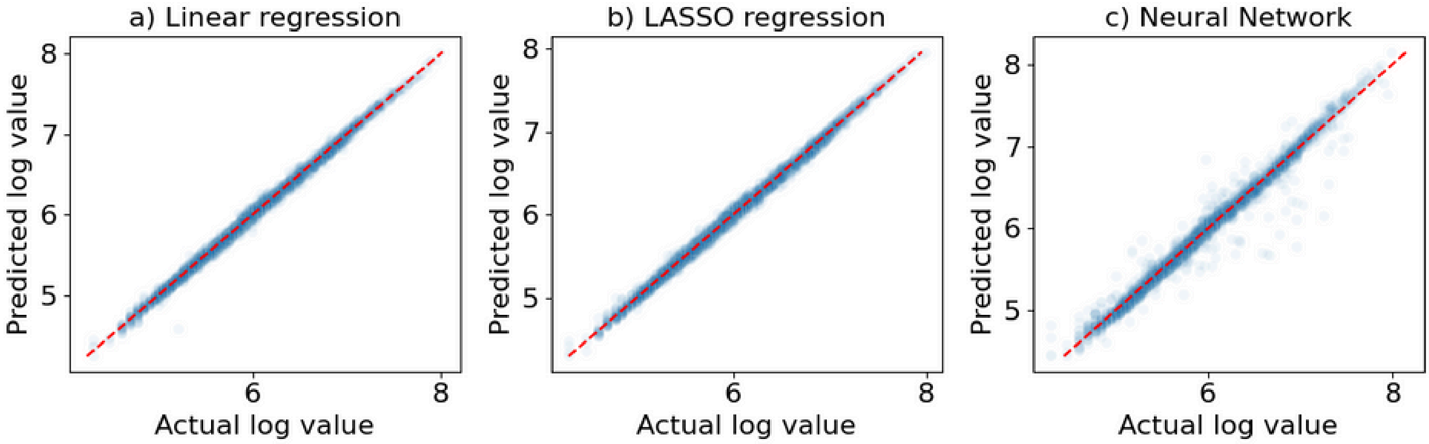


Figure 4 – Scatterplot of predicted versus actual log values of age for (a) *Linear regression*, (b) *LASSO regression* and (c) *Neural Network*

Model	Train RMSE	Test RMSE
Linear regression	0.034	0.037
LASSO regression	0.039	0.039
Neural Network	0.071	0.086

Table 2 – Summary of train RMSE and test RMSE (for log value)

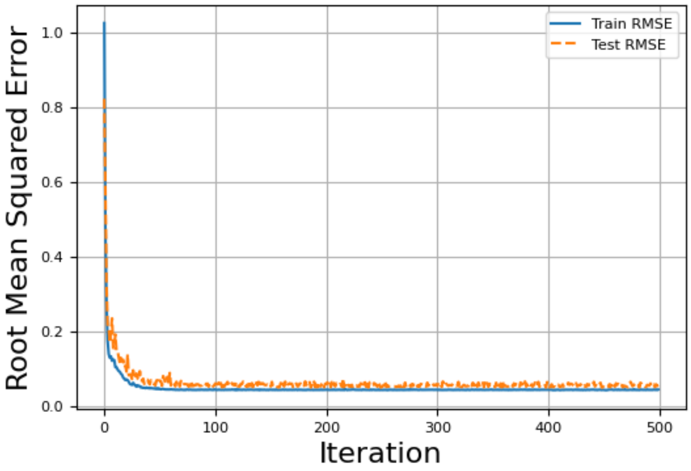
4.2.1. Linear regression

The RMSE of this model is 0.034 on the training set, and increase slightly to 0.037 on the testing set (see **Figure 4.a** and **Table 2**). This tiny gap suggests very low variance. The very low value of test RMSE (0.6% of averrage log value) indicates that the model do not suffer from biais.

4.2.2. LASSO regression

The RMSE of this model is 0.039 on the training set, and increase to 0.039 on the testing set (optimized 1 / λ = 0.001, see **Figure 4.b** and **Table 2**). There is almost no gap, so the varience is insignificant. The very low value of test RMSE (0.6% of averrage log value) indicates that the model do not suffer from biais.

4.2.3. Neural Network



The RMSE of this model is 0.071 on the training set, and increase to 0.086 on the testing set (see **Figure 4.c** and **Table 2**, see also **Table A2** for optimized hyperparameters). This small gap suggests medium variance. The low value of test and train RMSE (less than 1.4% of averrage log value) indicates that the model suffer from a little bias. The error curve show an optimum around 50 iterations (**Figure 5**) and stay at the same value, indicating that the model minimize error with an early stopping. Test RMSE curve follow Test RMSE curve, it means that model do not overfit, and work well with new data.

Figure 5 – Error curve of the neural network

5. Conclusion

The best model in term of test RMSE is **Linear regression** in every cases, with a test RMSE of 493,034 for value prediction, and 0.037 for log value prediction. To compare value and log value prediction, we have to apply a 10^x function to log value RMSE : $10^{0.037}$ = 1.088€

To conclude, predicting log value then applying inverse function allows us to predict cost with an error of 1.0€ ($4x10^5$% of the averrage market value) instead of 493,034.0€, so we will prefere this method.

6. Annexe

Hyperparameter	Tested values	Optimized values
Hidden layer sizes	[1], [2], [2,2], [4,2], [8,4], [16,8]	[4, 2]
Activation function	ReLU, Tanh, Logistic	ReLU
Solver	adam, sgd	adam
Initial learning rate	0.001, 0.01	0.01
Batch sizes	32, 64	64

Table A1 – Summary of train RMSE and test RMSE (for log value)

Hyperparameter	Tested values	Optimized values
Hidden layer sizes	[1], [2], [2,2], [4,2], [8,4], [16,8]	[8, 4]
Activation function	ReLU, Tanh, Logistic	ReLU
Solver	adam, sgd	adam
Initial learning rate	0.001, 0.01	0.01
Batch sizes	32, 64	32

Table A2 – Summary of train RMSE and test RMSE (for log value)