Who's Overpaid? Predicting Player Value Using FIFA Data

Rye Julson, Nico Manzonelli, Jeffery Mayolo, Joseph Zuccarelli

December 13, 2021

1 Introduction

FIFA is a popular football simulation video game produced by the company EA Sports. Every year EA Sports releases a new version of the game that includes the real-life players on each team. Each player included in the game comes with a large set of attributes concerning their physical features, athletic ability and skill level. For instance, a few popular attributes are height, weight, pace, shooting, passing, dribbling, defending and physical. In real-life, each FIFA player is payed a specific salary value by team ownership based on their production level on the field. However, is it possible that given the large amount of FIFA data concerning each player, we can build a model that provides better estimates of player salaries than human owners? **In the following study we will attempt to predict player value from player physical, athletic, and skill attributes using various machine learning models trained on publicly available data from the video game FIFA.**

2 Data

2.1 Data Collection

The website "sofifa.com" provides relevant and updated FIFA data for each player. After identifying several interesting quantitative and categorical variables for further exploration, we web scraped via BeautifulSoup to extract 12,000 relevant player variables. We focus on the following variables: player ID, name, origin country, club, primary position, age, height, weight, primary foot, value, attacking, skill, movement, power, mentality, and defending. Player ID is a unique identifier for each player. Our response variable, player value, is the salary amount for each player in euros. Our predictor variables – attacking, skill, movement, power, mentality, defending – represent numerical scores for each player attribute.

After collecting the data, we shift our focus to cleaning the data in order to enable for further exploration. First, we remove any repeated rows based on player ID. This narrows our data set to 11,915 unique athletes. Note that we also remove goal keepers from our analysis because they are not evaluated using the same metrics as field players. This narrows our data set to 10,792 athletes. Next, we properly transform all quantitative predictors to integers, as all observations on "sofifa.com" are listed as characters. Lastly, we confirm that our data set does not include any missing values.

2.2 Data Exploration

First, we examine the summary statistics for all the variables included in our data set (see Table 1). Notice that the distributions of all the variables included in our data set appear to be

relatively symmetric based off the mean and median values, except for defending and player value.

Table 1: Numerical Summary

	Age	**Height (cm)**	**Weight (kg)**	**Attacking**	**Skill**	**Movement**	**Power**	**Mentality**	**Defending**	**Player Value (euros)**
Mean	26.03	180.58	74.87	289.23	298.66	340.05	325.98	291.30	162.86	4.32×10^6
Median	26.00	180.00	75.00	291.00	302.00	345.00	327.00	291.00	186.00	1.70×10^6
STD	4.17	6.57	6.73	40.79	48.66	43.44	33.37	35.86	54.03	9.30×10^6
Min	16.00	156.00	53.00	165.00	140.00	168.00	184.00	180.00	35.00	0.00×10^0
Max	42.00	203.00	110.00	437.00	470.00	461.00	444.00	414.00	272.00	1.86×10^8

Next, we refine our exploration to the response variable, player value. Figure 1 included below displays the distribution of player value for all the observations included in our data set. Notice the strong right skew in the distribution of the normal response in the left boxplot. Therefore, we use a log transformation (base 10) to transform the response variable, as this allows us to validate the normality assumption of the regression models that we describe in the following section. The right plot included in Figure 1 shows the distribution of the log transformed response variable. Notice that this distribution appears to be much more normal with a few outliers on the right.

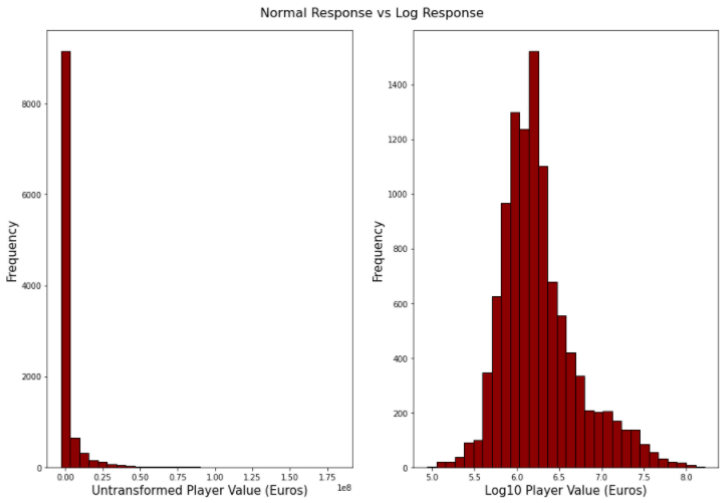


Figure 1: Distribution of Player Value (Regular Scale vs. Log Scale)

Finally, we explore the relationship between the response variable, player value, and a few of the predictor variables included in our data set. Figure 2 included on the following page displays the relationship between player value and the following variables: age, height, weight, attacking, skill, movement, power, mentality, and defending. Notice that there appears to be a positive relationship between player value and the various player attributes, as expected.

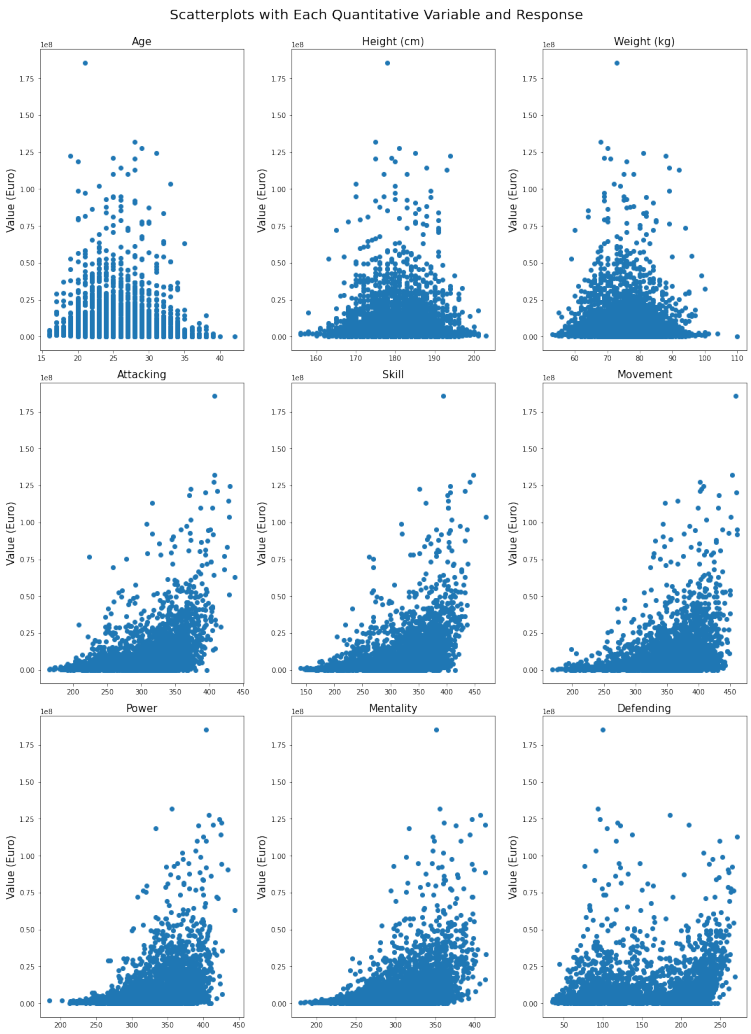


Figure 2: Player Value vs. Predictor Variables

3 Methodology

After exploring our data set, we fit several different machine learning models to predict player value from various player physical, athletic, and skill attributes. Prior to fitting any of the models outlined below, we split the data set using an 80-20 train-test split. All model fitting and cross-validation is performed on the training set – the test is only used to report the final accuracy of our chosen model.

First, we fit a multi-linear regression model with all predictors in our data set to serve as a baseline. This model is structured as follows:

$$\log_{10}(\hat{Value}) = \beta_0 + \beta_1 Age + \beta_2 Height + \beta_3 Weight + \beta_4 Foot + \beta_5 Attacking + \beta_6 Skill + \beta_7 Movement + \beta_8 Power + \beta_9 Mentality + \beta_{10} Defending$$

Second, we fit a k-Nearest Neighbors (KNN) model to predict player value. This requires normalizing all model inputs prior to fitting. We use 10-fold cross-validation to tune our main

model hyperparameter– the number of nearest neighbors (k). In performing this cross-validation we iterate over eight different values for k (1,2,3,5,7,10,50,100). We select the k value that results in the lowest average cross-validation MSE. To adjust for potential overfitting to the train and validation sets, we perform bootstraped sampling to reassess the most desirable k value.

Third, we fit a regularized (LASSO) polynomial regression model to predict player value. We do this as we hypothesize many of the predictors have non-linear relationships, as displayed by the scatterplots in Figure 2. We initially transform the data to include up to the cubed term for each predictor. Then, we allow LASSO regularization to perform feature selection. Effectively, this will shrink the coefficient terms of non-important variables to 0, reducing model complexity. We use 5-fold cross-validation to tune our main model hyperparameter– the shrinkage parameter (α). In performing this cross-validation, we iterate over six different values for α ($10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, .01, 1$). We select the α value that results in the lowest average cross-validation MSE.

Fourth, we fit a few tree-based methods to predict player value. First, we fit a single decision tree regressor. We use 5-fold cross validation to tune our main model hyperparameter– max tree depth. In performing this cross-validation, we iterate over tree depths 1-20 (inclusive). We select the tree-depth that results in the lowest average cross-validation MSE. Next, we fit a random forest regressor with 50 trees. We use 5-fold cross-validation again to determine the optimal max depth of each tree (i.e. the max tree depth that results in the lowest average cross-validation MSE). In performing this cross-validation, we iterate over several max tree depths, many of which are overfit. The tree depths we iterate over are 5, 10, and 15-20 (inclusive). Lastly, we fit an AdaBoost regressor. First, after doing some exploration, we determined performance did not increase with more than 200 estimators. Therefore, and since the number of estimators plays a vital role in computational load and performance, we set the maximum to 200. Then, we cross-validate over several base tree depths (5, 10, 15, and 20) with this maximum number of estimators. We select the base tree depth that leads to the lowest validation MSE.

Fifth and finally, we fit a simple TensorFlow neural network model. Due to the complexity of these models, we do not perform tuning or cross-validation. Instead, we train based on general convention with 5 layers, each with 'relu' activation functions, and a decreasing number of nodes (64, 32, 16, 4, 1).

After tuning all the machine learning models described above, we compare their performance in terms of validation set MSE. We select our best model based on the lowest validation MSE.

4 Results

The modeling approach described in the previous section led to the following results. First, we fit our baseline linear regression model on the entire training data set. Note that this did not require us to perform any cross-validation, as this model does not include any hyperparameters that require tuning. This model is not to be considered for selection as a best predictive model–its intended purpose is strictly for comparison.

Next, we tuned the KNN model based on the number of nearest neighbors– k. Figure 3 displays the training and cross-validated MSE values at various values of k.

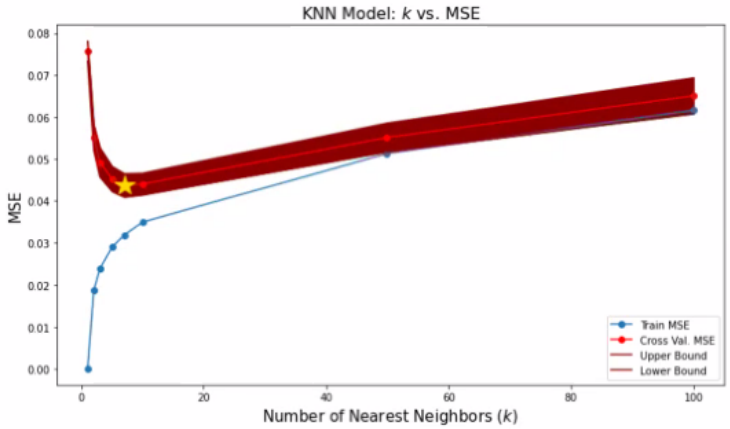


Figure 3: KNN Model Results

As illustrated by Figure 3, the k that achieved the lowest cross- validated MSE was a k of 7. However, to adjust for over-fitting, we performed a bootstrap sampling technique to determine how many times each k value is chosen through cross-validation. Through bootstrapping, we noticed that a k of 5 is selected the most on boostraped samples of the train data. Therefore, the ideal KNN model uses a k of 5.

Moving forward, we performed cross validation to tune the L2 penalization term in polynomial LASSO Regression: α. Figure 4 displays the train and cross-validated MSE's for various shrinkage parameters (α).

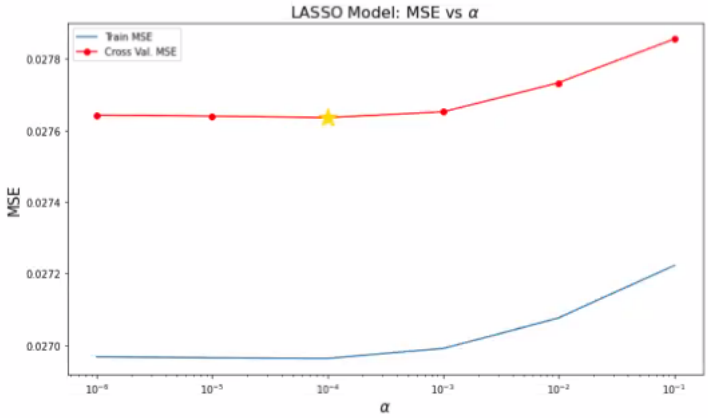
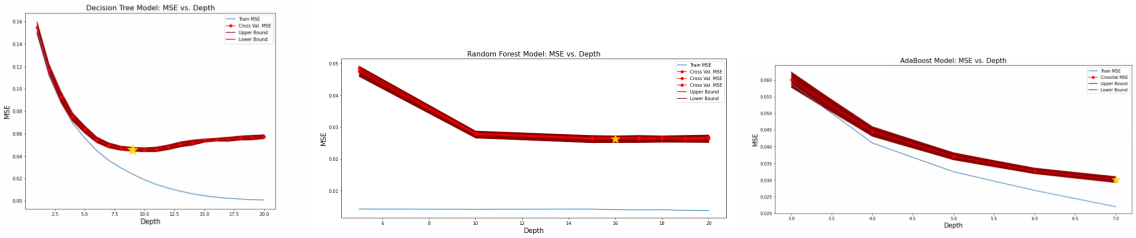


Figure 4: LASSO Model Results

Figure 4 reveals that using an α of 0.0001 yields the lowest cross-validated MSE. Therefore, we select 0.001 as the L2 regularization term to use for LASSO regression when comparing validation MSE's for final model selection.

Next, we implemented some non-parametric decision tree based prediction models and tuned on tree-depth when fitting a single decision tree, a random forest, and an AdaBoost model. Figure 5 displays the train and cross-validated MSE's for each model at various depths.



(a) Single Tree Model (b) Random Forest Model (c) AdaBoost Model

Figure 5: Tree Based Methods Results

As evident by figure 5 (a), the single decision tree with depth of 9 achieves the lowest cross-validated MSE. Due to the stochastic nature of random forest models, we can fit deeper trees which will be aggregated together in order to reduce variance. Therefore, as shown above, we see that cross-validation selects **16** as the optimal tree-depth for this type of model. Next, we move to AdaBoost. With this method, instead of aggregating deeper trees together, it cyclically combines smaller trees (or weak learners) based off their residuals. We see that **7** trees has the lowest cross-validation MSE.

Lastly, using tensorflow we built a simple neural network for regression. Trained over 100 epochs, we achieved promising validation-set performance from the neural network, especially considering that we trained the network on the simple non-polynomial inputs. However, due to the complexity of neural nets and the abundance of hyper-parameters to tune (activation functions, hidden layers, optimizers, etc.), creating the optimal neural net is beyond the scope of this study. Therefore, we leave the neural network results as is.

Based on the train and validation MSE's we can select the best predictive model out of many well-tuned models. In the end, we see that the Random Forest model achieves the lowest validation score of 0.0263. However, the extremely low associated train MSE leads us to believe that this model may be overfit to the training set. Instead, the well-tuned LASSO regression model achieves a validation MSE of 0.0266 (only 0.003 higher), and provides a more reasonable train MSE. Therefore, we think the LASSO model will generalize best to other tests, and we select it as our final model.

Table 2: Model Results

Model	**Train MSE**	**Validation MSE**
Linear Regression (Baseline)	0.0883	0.0885
KNN ($k = 5$)	0.029	0.0452
LASSO ($\alpha = 0.0001$)	0.0258	0.0266
Single Tree ($depth = 9$)	0.0297	0.0455
Random Forest ($depth = 16$)	0.0051	0.0263
AdaBoost ($depth = 7$)	0.0224	0.0292
TF Neural Net	0.0319	0.0338

Finally, based on the model results and general intuition we conclude that polynomial LASSO regression is the best model to capture the overall trend in player value. We evaluated the final LASSO model on the test set and achieved an unsurprisingly low test MSE of **0.0287** with an R^2 value of **0.8578**. This informs us that the LASSO model explains **85.78%** of the variation in the log (base 10) of player value.

5 Discussion

By only considering physical traits and measures for athleticism, we aimed to model the relationship between player value and athletic ability. Intuitively, it makes sense that the most athletic players should be payed the most. However, exploring how certain clubs value their athletes may shed light on player evaluation. To determine if a player is over payed or under payed we compare the model's predicted value to the actual value for the player. If the difference in model prediction and actual value evaluation is positive, our expected value is greater than the actual value and we conclude that the athlete in question is over payed. The inverse applies to determine if an athlete is under payed. If the difference in model prediction and actual value evaluation is negative, our expected value is less than the actual value and we conclude that the athlete in question is under payed. Figure 6 contains the top 10 undervalued players and top 10 overvalued athletes in the entire FIFA dataset.

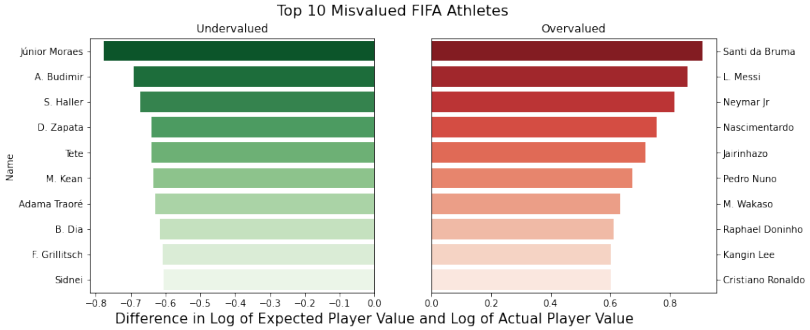


Figure 6: Top 10 Misvalued Athletes

As expected, our model considers many lesser known athletes as undervalued. Logically, we expect many of the undervalued players to be unknown which indicates that there are good 'budget' footballers. If we were using this model to build the roster of a small-market football club, these are the players we would target first. We also classify many lesser known athletes as 'overvalued', which provides evidence that their clubs may be over valuing them. Interestingly, many internationally renowned footballers as overvalued. Stars like Messi, Neymar, and Ronaldo have instant name recognition. These players may be considered overvalued because their actual value is so high they represent outliers in the data set. However, their classification as 'overvalued' reveals a key modeling limitation: we only account for athleticism via proxy FIFA data. The raw ability to score game winning goals under pressure and significantly impact a football match (the reason clubs pay their star athletes so much) is not adequately captured in the model.

Using a similar process to evaluate each individual athlete value, we can assess how each club values their athletes by comparing the actual value of the average player each the club against the predicted value of the average player each the club. Figure 6 displays the top 10 clubs that misevaluate their athletes.

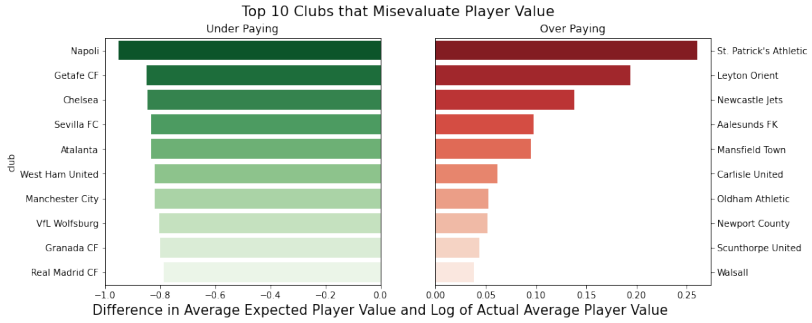


Figure 7: Top 10 Clubs that Misevaluate Athletes

Although we expected many 'big name' clubs with large budgets (think PSG, Barcelona, Real Madrid, or Juventes level) to be some of the most over paying clubs according to our model, instead some clubs with name recognition appear to be undervaluing their athletes. While these clubs to have relatively high budgets and could hypothetically afford to overpay their athletes, it appears that these clubs actually pay significantly less for their average player than the model expects. Instead, many small clubs appear to be overpaying. We expect that because there is a strict floor in talent for professional footballers (even the lowest payed professional athletes are still phenomenal athletes) smaller clubs suffer from having to overpay athletes.

Although our results provide interesting insights into predicting player value, the models developed in this study do contain a few limitations. One important limitation is that the model only predicts player value based on 5 key FIFA player attributes and a few other physical attributes. Instead, we know that players are valued for many different reasons, and although the FIFA data collected is meant to serve as a proxy for player common player evaluation metrics, it's likely this doesn't fully encompass all factors that determine a players worth. Additionally, the results are not vert generalizable. For example, our model could not extend to other sports and is only based on one year of FIFA player data.

6 Future Work

Given the limitations outlined in the previous section as well as the time constraints imposed on our study, there are several opportunities for future research. First, future researchers should attempt to use the modeling approach outlined in this study to predict player value within other sports, as other video games such as Madden and MLB the Show posses similar data concerning their real-life players. Such a study would perhaps provide insight in terms of the generalizability of our chosen model. Second, future studies should analyze FIFA player data across several years to gain insight concerning how players' value changes over time. Finally, future researchers should explore how our model may introduce potential bias in player value evaluation based on race or nationality, or explore how to correct for bias in player value based on race or nationality at a higher level.