Football Player Performance Analysis using Particle Swarm Optimization and Player Value Calculation using Regression

To cite this article: Ayanabha Jana and Sweetlin Hemalatha 2021 *J. Phys.: Conf. Ser.* **1911** 012011

Football Player Performance Analysis using Particle Swarm Optimization and Player Value Calculation using Regression

Ayanabha Jana and Sweetlin Hemalatha

Vellore Institute of Technology, Tamil Nadu 600127, Chennai, India

Email: ayanabhajana@gmail.com

Abstract. This paper aims at analysing the statistical data of various football players to establish a correlation between their play style and individual scores with their quantifiable attributes. Having established a correlation, the next step involves using the Particle Swarm Optimization(PSO) to simulate a match and draw a comparison between two random players, constraining their attribute scores within the boundaries of the top-recommended player for each attribute, as suggested by the k-nearest neighbors algorithm. This aids in setting up a benchmark score for the particular player position for a random selection from this subset based on z-score inference. Having optimized the player position score, stepwise regression and smoothing splines are used to model a prediction to compute the overall score of the player. Lastly, a regression equation is modelled using stepwise regression to estimate the net worth of the player based on their skill set, and predictions are performed using the optimal score obtained from PSO, by extracting the individual attribute scores from the inverse regression relation. From the experiment, the optimized score for the left striker(LS) comes out to be 86.32766. Running the PSO on all left strikers gives a 98% probability of obtaining a player whose score is greater than the benchmark score. For the two left strikers whose scores were optimized, the predicted worth from the stepwise model comes out to be 8.933734 and 8.191562, the former being greater than the historical worth.

1. Introduction

The analytics performed on dataset [1] in this paper falls under two categories:

- On-field analytics – correlations are computed between the various player skills and the player positions which are used to build a regression model that estimates the player position score based on the positively correlated attributes. A particular player position is then selected to be optimized using PSO. Once all the player scores for this position are optimized, a player is selected at random to check if it satisfies the benchmark score for that particular position. Having acquired the player position scores, the player overall scores are calculated using regression and spline modeling.
- Off-field analytics – a regression model is built to calculate the player worth based on the player attributes. Inverting the results of the optimized player position score estimated before using PSO gives these attribute values which when plugged in the regression model gives the predicted player worth.

2. Literature Survey

Player Performance Prediction in Football Game [2] utilizes a supervised learning algorithm to predict the overall performance and market value of the player. Using principal component analysis, a set of uncorrelated player attributes is obtained to compute the overall performance of the players segregated into different classes like Attack, Goalkeepers, Midfielders, etc. Considering a feature vector of significant attributes for players of different classes, a linear regression model implemented on the training set is used to predict the overall performance of the testing set. The next step involves the usage of a regression model to estimate the market value of the player based on their overall performance. A log transform of the dependent variable is used to tally with the exponential nature of the market value against the overall performance. The performance model gives an accuracy of 84.34%, with a R-square value of 0.84 while the market value model gives an accuracy of around 91% on the testing set, with a R-square value of 0.91.

Football Player's Performance and Market Value [3] utilizes regression trees to estimate the *Market Value* of the players for various teams and nationalities but since the chosen dataset has more variables than observations, the results obtained are too general, and hence lasso regression is employed to perform variable selection in the linear model with a better accuracy that minimizes the usual sum of squared errors. However, using λ_{min} as the threshold for lasso regression gives a negative market value while using λ_{1se} restricts the usage of predictor variables. Hence the value $\min(\lambda_{min} - \lambda)$ is used whenever the predicted market value turns out to be negative. The model for market value thus formed is:

$$\hat{M} = 231.26 + 0.89.Market.value + 2723.36.Assists \tag{1}$$

[3] next segregates the player in the chosen dataset into 4 classes – Goalkeepers, Defenders, Midfielders and Strikers and utilizes t-test to check for significant difference between the classes in terms of predicted market value acquiring a p-value of 0.001 and for subclasses within each class a p-value of 0.8, suggesting from the perspective of market value, more than four classes of players are not justified. Using the forward players' data, lasso regression together with five-fold cross-validation gives a model for player performance as follows:

$$\hat{P} = 0.28 - 0.073.F + 0.06.SP + 0.04.ST + 0.02.GP + 0.05.GB + 0.02.D + 0.08.A \tag{2}$$

The models for market value and player performance in [3] are now studied to analyse the underlying trend between these two parameters, which suggests a directly proportional relationship. The difference between the real and predicted market value calculated as $\Delta = \ln(M) - \ln(\hat{M})$ gives the accuracy of the calculated market value with a lower Δ indicating higher accuracy. For $\Delta > 0.3$ the player is over-valued and if $\Delta < -0.3$, the player is under-valued.

[4] focusses on predicting the performance of football defender players' performance by employing a multiple regression model with backward elimination by splitting the dataset features into three categories – player characteristics and attributes, player statistics, and team statistics. For the first category of features, the Durbin-Watson test gives a value much lower than 2, which implies a positive autocorrelation between features and the R-squared and adjusted R-squared lower than 0.5. The second category of features gave a model with twelve features with a much better R-squared and adjusted R-squared value of 0.867 and 0.833 respectively. The only feature from the third category that increased the R-squared value to 0.907 and the adjusted R-squared to 0.88 of the previous model is 'TeamRating'. Some of the important features that contribute to the defensive players' performance include interceptions, jumping reach, strength and passing.

[5] uses stepwise regression with backward elimination to predict the variables related to athletic performance based on physical and hematological parameters like blood samples, heavy clothing, age, BMI, etc. The five response variables related to performance are Countermovement Jump (CMJ), Ten Meter Sprint, Aerobic Evaluation (VO_2max), RSA Total Time and RSA Index, having the final number of predictors as 5,5,9,4 and 5 respectively. Each of the dependent variables having an adjusted R-squared of more than 0.5 indicates the goodness of fit and lower standard errors of the independent variables as shown in table 2 of [5]. [5] effectively reduces the number of predictors for estimating athletic performance and this aspect of stepwise regression will be used in the forthcoming sections to

predict the overall score and player worth but with an understanding of the complexity reduction in stepwise regression.

One shortcoming of [2] and [3], is that the player attributes are bound to change and such variations change the overall performance and market value of the player i.e. both [2] and [3] describe models that operate on historical data and can only give predictions without implementing any scope of improvement. [4] on the other hand, specifies the performance of only defenders and estimates player performance on basis of static data only. In reality, however, attribute scores change after every match and performance prediction methodology should apply to all categories of player positions. A better approach would be to see what level of optimization can be obtained for a particular class of players via these attributes to tally with their historical data and using this optimized value to calculate the performance and the market value of the player. This comparison can help in identifying the improvement needed for a particular player, as discussed in the upcoming sections.

3. Diagnosing Player Attributes and Play Styles

3.1. Correlation between attributes and play styles

This project utilizes the Fifa19 dataset [1]. The first step towards understanding a player's performance would be to estimate how their overall or individual play styles vary according to their skill set. Correlation analysis has been employed to gain a visual insight on the strength of these skills, or attributes inclined towards a particular play style, or position.

A glimpse of these correlations is depicted in a tiled heat-map shown in figure 1. The light-blue tiles indicate a strong positive correlation between the skill (vertical axis) and player position (horizontal axis), while the darker shades represent strong negative correlations. This indicates that not all attributes contribute equally in determining the player position scores. Table 1 gives the expanded form of the various player positions.

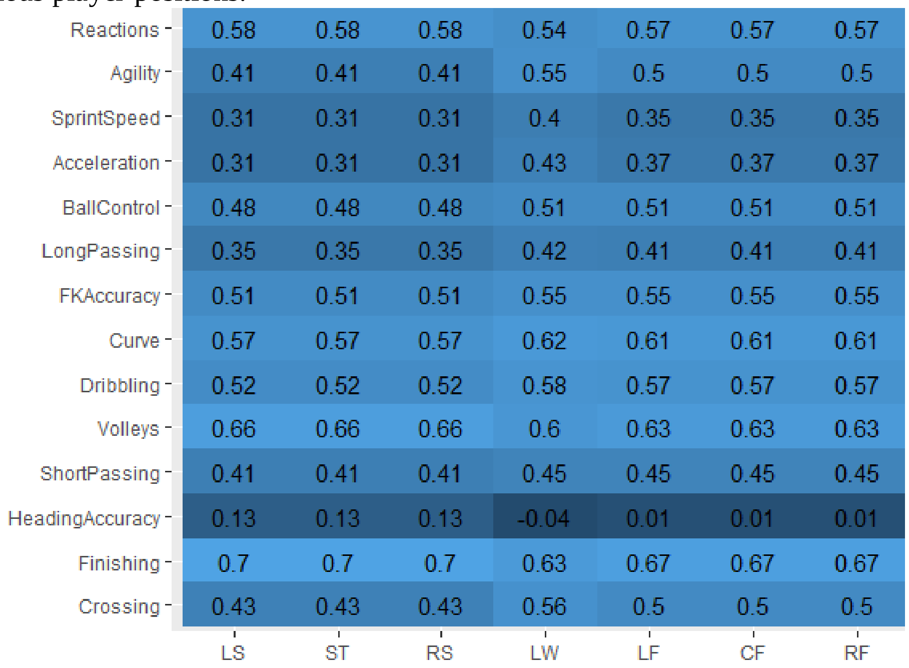

Figure 1. Heat-map resembling correlation between player attributes and positions

Table 1. Expanded forms of various player positions

Abbreviation	Expansion	Abbreviation	Expansion
LS	Left Striker	LF	Left Forward

ST	Striker	CF	Central Forward
RS	Right Striker	RF	Right Forward
RW	Left Winger		

4. Simulating a match using PSO algorithm

4.1. Particle Swarm Optimization

Particle Swarm Optimization or PSO is an iterative algorithm that helps to achieve the optimal solution i.e. either maximum or minimum possible value for a given problem by constantly updating the search space under certain constraints starting from an initial position and velocity.

Consider the maximum possible score that is obtainable by a left striker (LS) needs to be found. Having filtered only the left strikers from the dataset, a multilinear regression model is devised from the results obtained in Section 3.1 with the independent variables set to the ones with the highest positive correlation. A summary of the regression model is as follows:

```
Call:
lm(formula = LS ~ Finishing + Volleys + Curve + Reactions + LongShots +
    Positioning + Vision, data = dmatch)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8775 -0.9972  0.0191  1.0250  4.9380

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.52266    1.01921  13.268  < 2e-16 ***
Finishing    0.33151    0.02731  12.141  < 2e-16 ***
Volleys      0.06983    0.01937   3.605 0.000395 ***
Curve        0.00442    0.01113   0.397 0.691707
Reactions    0.09210    0.02710   3.399 0.000818 ***
LongShots    0.06927    0.02148   3.226 0.001470 **
Positioning  0.20562    0.03079   6.679 2.35e-10 ***
Vision       0.03054    0.01536   1.988 0.048151 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.51 on 199 degrees of freedom
Multiple R-squared:  0.9424,Adjusted R-squared:  0.9404
F-statistic: 465.3 on 7 and 199 DF,  p-value: < 2.2e-16
```

From the above results, the values of estimates are used to frame the equation for calculating the left striker score as follows:

$$LS{=}0.33151 * \mathit{Finishing} + 0.06983 * \mathit{Volleys} + 0.00442 * \mathit{Curve} + 0.09210 * \mathit{Reactions} + 0.06927 * \mathit{LongShots} + 0.20562 * \mathit{Positioning} + 0.03054 * \mathit{Vision} + 13.52266 \tag{3}$$

Equation (3) is the problem function that needs to be optimized using PSO [6]. Considering the players as individual particles the maximum particle velocity is set to the maximum value of the player

‘Agility’ column normalized by 100. The inertia weight w and the constants c1 and c2 are set to the default values of 0.7, 1.5 and 1.5. The PSO algorithm in this case deals with 7 variables whose boundaries are set to the maximum and minimum values of their respective columns. Running the algorithm for 100 iterations and 20 particles, the maximum left striker score that is obtained is:

```
## [1] 86.32766
```

4.2. Recommendation using k-nearest neighbors

In section 4.1, the maximum possible score for all the left strikers was obtained using the default configuration of PSO. However, in reality, it would be much more suiting to achieve optimization under the most constraining of conditions. The idea is to limit the range of the search space variables i.e. the player attributes on which the player position score depends by setting their bounds to the values of the top-recommended player for that particular attribute, which is analogous to the player under analysis facing off against the best players holding a specific skill.

The KNN algorithm is used to build a simple recommendation system that takes a distance matrix between the player attributes as an input along with a skill id and returns the first k nearest neighbors for that particular id, indicating the top players for a given attribute or skill. KNN does this by calculating the distance between the target player and the other players and ranking them as per the ones which are nearer to the target. Figure 2 shows the top 5 players for the skill ‘Crossing’ along with a comparison of their potential and the overall score, from which it is evident that for a given skill, the potential and overall score are almost similar for the nearest neighbors.

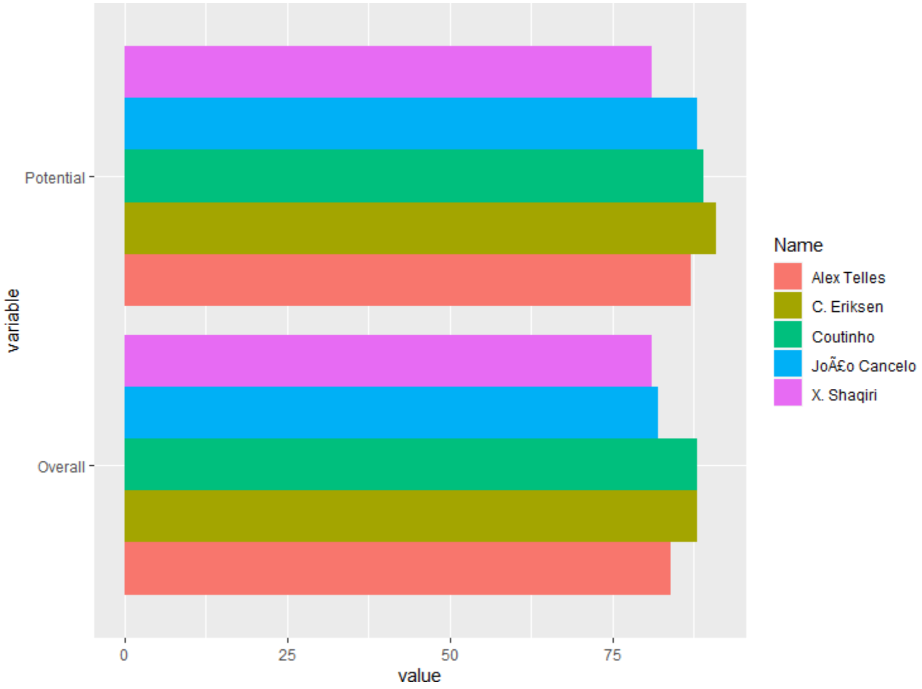

Figure 2. Top 5 players for ‘Crossing’ Skill

4.2.1. Using top players’ data in PSO. Using this recommender, the top player for each skill that contributes to the left striker score namely, ‘Finishing’, ‘Volleys’, ‘Curve’, ‘Reactions’, ‘LongShots’ and ‘Vision’ are found. Now two left strikers are randomly selected whose scores are to be optimized. The PSO algorithm [6] is run separately for these two players on equation (3) with the parameters set as follows:

$$V\_Max = (player\ agility)/100 \tag{4}$$

$$c1 = 1.5,\ c2 = 1.5 \tag{5}$$

$$w = (\mathit{player\ weight})/200 \tag{6}$$

where,
c1 = individual acceleration for PSO, c2= group acceleration for PSO
w = inertia weight for PSO

The upper or lower bounds on the search space variables (or, player attributes) are the attribute value of the top player for each skill, with a particle size of 11, as in 11 opposing players in a football match and a total of 100 iterations. The optimized score for the first and second player reported as the average of the PSO result and historical value, are as follows:

```
## [1] 74.49339

## [1] 72.99339
```

Figure 3 shows a comparison of players' historical scores and the ones obtained from the match simulation. The simulation values are greater than the historical values due to the maximization aim. Lesser values are obtainable when the higher bound of the search space is the player position score itself.

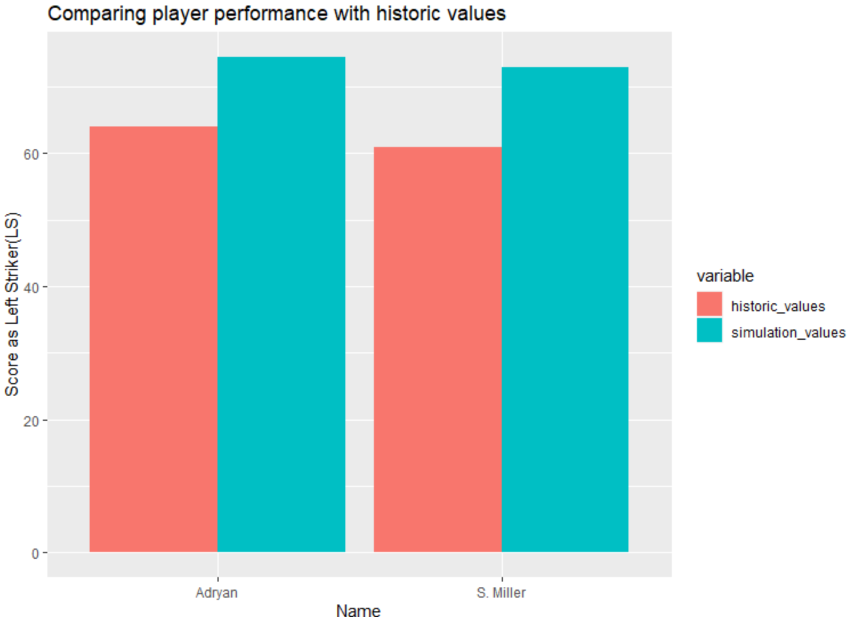

Figure 3. Comparison of left striker historical score vs simulation score

4.3. *Randomized Player Selection*

Using the same constraints as defined in section 4.2, the PSO match simulation [6] can be executed for all the left-strikers to get their optimal scores. Based on this result, the left-strikers can be selected randomly for making the team. But, given a random selection how one is supposed to know that the selected score is satisfactory to contribute to the overall performance of a team in a match? One can select the player with the highest optimal score but that would just leave out the other players from the selection process in each turn.

A better approach would be to set a benchmark score to which a randomly selected score would be compared. Rather than selecting the highest score, one can just check if the selected score falls above or below the benchmark and compute the probability of the same. This approach can be implemented using z-score inference on the optimal scores obtained from the PSO algorithm. Start by assuring a normal distribution of the optimal scores from figure 4, which gives an approximate bell-shaped curve.

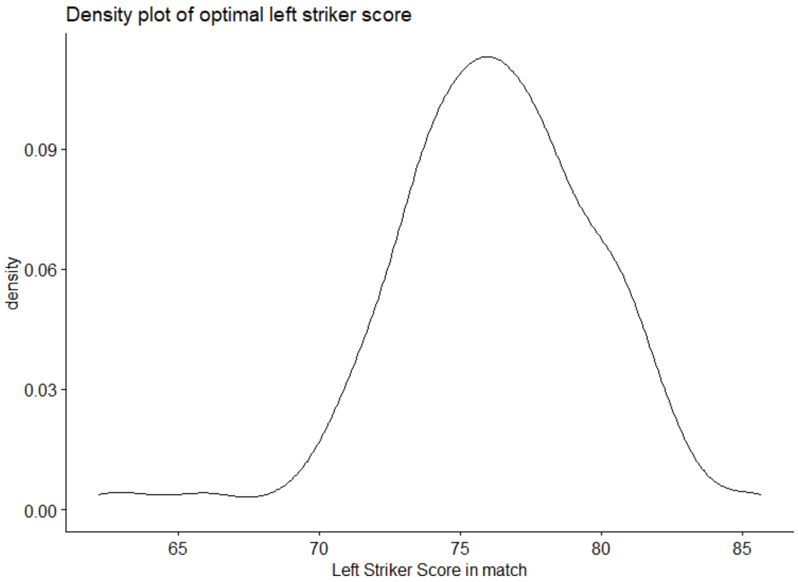

Figure 4. Density plot of optimal left striker score for checking normal distribution

Now, assume the benchmark score to be the left striker score obtained from the attribute scores of the top players and set the null hypothesis and alternative hypothesis as follows:

H0 : Player score <= Benchmark score

H1: Player score >Benchmark score

The z-score is then computed as follows:

$$z = (x-\mu)/\sigma \tag{7}$$

where,

x = benchmark score

μ = mean of optimal scores

σ = standard deviation of optimal scores

Following are the results of the z-score and corresponding probability.

```
> mu<-mean(dmatch$optimal_LS)
> s<-sd(dmatch$optimal_LS)
> z_score<-(benchmark_LS-mu)/s
> z_score
[1] 2.394923
> prob1<-2*pnorm(-abs(z_score))
> prob1
[1] 0.01662387
```

The probability here refers to the occurrence of the null hypothesis which is significantly lesser than the alternative hypothesis. Hence for a random player, he is most likely to satisfy the alternative hypothesis. From figure 5, one can see that there is a higher chance of occurrence of the alternative hypothesis, so whatever random player one selects there is a 98% chance that it will satisfy the benchmark score requirement and hence can be included in the team.

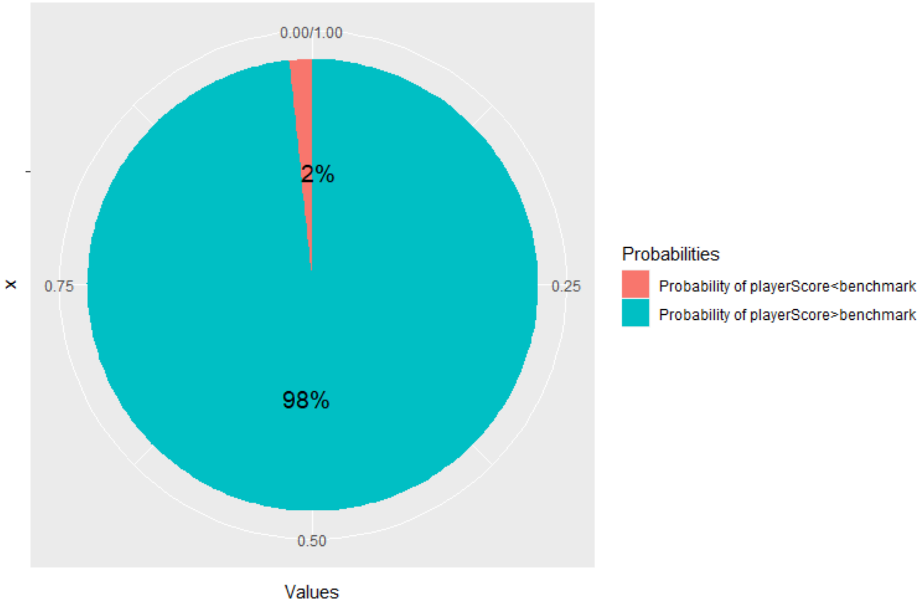

Figure 5. Probability of player position score greater than benchmark score

5. Estimating the overall score of a player

5.1. Using the optimal player position score

Now that the optimal player position score (in this case, Left Striker) has been obtained, one needs to compute the overall score of the player. In the chosen dataset [1], each player has a certain position score and an overall score. Assume that these position scores have a cumulative effect on the overall score, but not all of these need to have an effect.

A linear model is approximated for the overall score by using stepwise linear regression, which starts with all the player position scores under consideration and sets the lower bound as no independent variable. At each iteration, it computes a parameter known as the Akaike Information Criterion (AIC) and optimizes it at each turn by eliminating one or more independent variables, giving the final regression equation with the most correlation which is as follows:

```
## lm(formula = Overall ~ LS + LF + LAM + LDM + LB, data = dmatch)
```

where,

LS, LF, LAM, LDM, LB = Left Striker, Left Forward, Left Attacking Midfielder, Left Defensive Midfielder and Left Back scores

The complexity of the above model can be explained in terms of overfitting i.e. when the model fits perfectly to a set of data points, not considering the error in data entries leading to erratic predictions on test data as it starts learning from this error as well. Assuming the number of predictors to be proportional to the complexity, with more predictors the regression analysis is forced to find extra coefficients using the same sample size leading to erroneous estimates. Statistically, overfit models are detected by the difference between the actual R-squared and predicted R-squared as mentioned in [7]. If the difference is small, the model predicts the test data well i.e. it is less overfitted and therefore has less complexity. The following code calculates this difference for the overall score model of the player based on all the player position scores and the overall score model returned by stepwise regression:

```
> score_model_initial <- pred_r_squared(s_model2)
> initial_overfit <- abs(summary(s_model2)$adj.r.squared - score_model_init
ial)
> initial_overfit
[1] 0.002570643
```

```
>
> score_model_final <- pred_r_squared(score_model)
> final_overfit <- abs(summary(score_model)$adj.r.squared - score_model_fin
al)
> final_overfit
[1] 0.001674496
```

As is evident from the above result, stepwise regression gives a lesser difference and hence is less complex.

One thing to notice in the above result is that the historical value of LS is used to build the model. However, the optimal value of LS is used to predict the overall score of the two random players selected in section 4.2. Figure 6 gives a comparison of the actual overall scores with the predicted scores from the simulated optimal values. As is evident from figure 6, the predicted scores show a slight increase from their actual scores, due to maximization in PSO and the fact that the lower bound of the score is the player score itself. Hence any optimization will only exhibit a slight change in the scores.

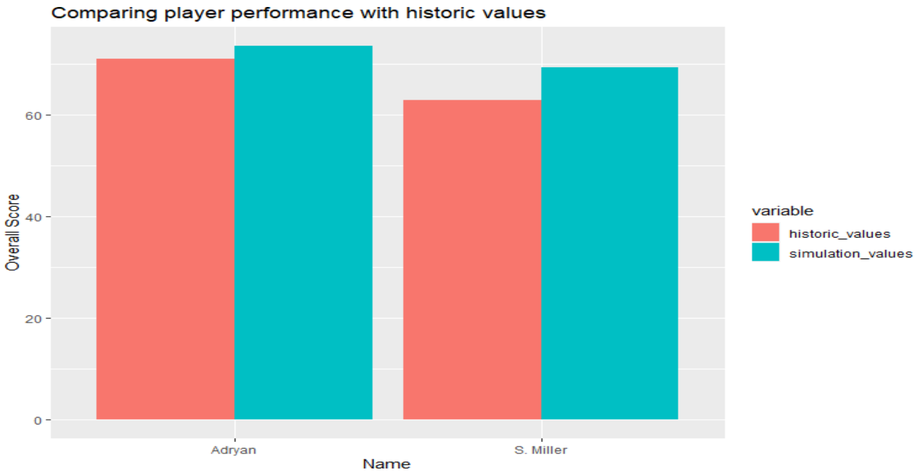

Figure 6. Player overall scores vs Predicted(simulated) scores

5.2. Smoothing Spline Estimates

The predicted scores or the simulation scores in section 5.1 are the overall scores of the players obtained from the results of the match simulation. Due to this, it can be assumed that there is some amount of unwanted noise in this data because these are very close to the actual scores. If one can just eliminate, or smooth out this noise or conversely introduce some amount of noise, one can obtain the simulated scores from the overall scores and vice versa.

This is obtained using smoothing splines which estimates a function to these noisy data (in this case, a cubic smoothing spline) by taking a vector of weights which is set to be the ‘Potential’ column for the players.

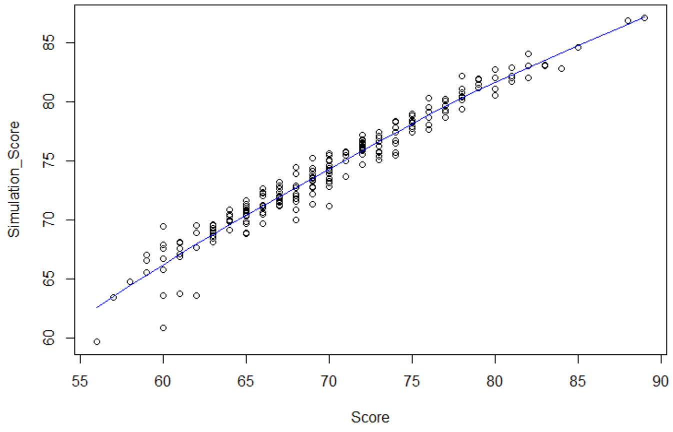

Figure 7. Smoothing spline for the overall scores

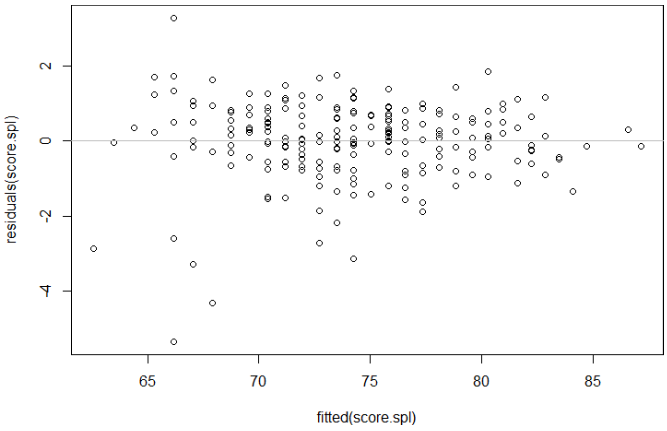

Figure 8. Residuals vs Fitted for actual and predicted overall scores

From figure 7, one can see an almost perfect fit for the simulated scores, which is not linear but a little curved due to the added effect of the spline weights, which reduces the overall residual error which would have been obtained from a regression model. One can verify this fit from figure 8, which has an almost horizontal line between the residuals and fitted values, which resembles a good fit. One advantage of using smoothing splines is that it gives approximately the same results as that acquired from regression in section 5.1 but with a much better fit as depicted in figure 9.

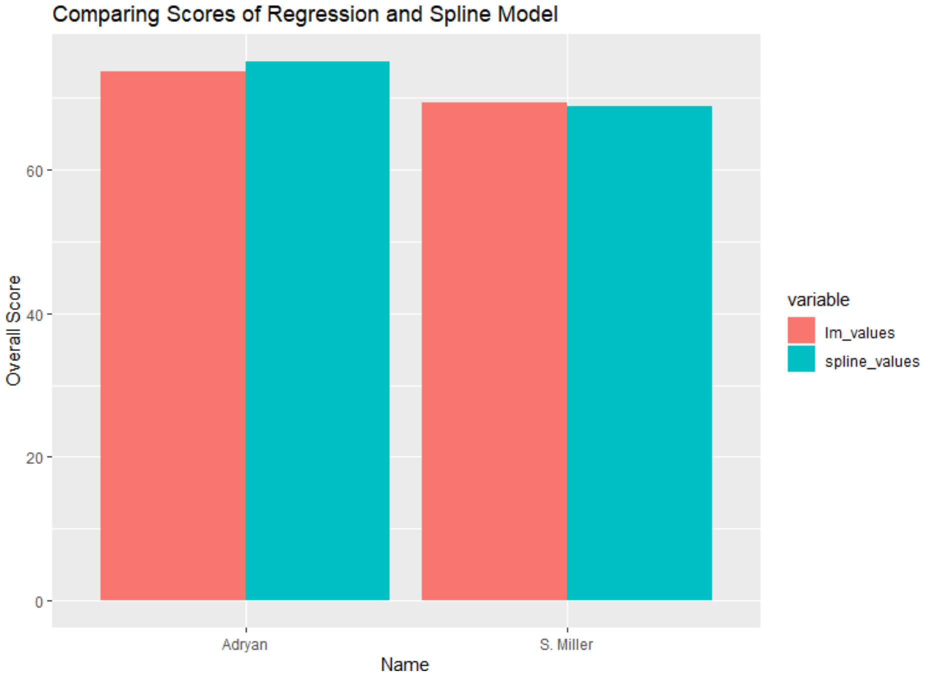

Figure 9. Predicted Scores obtained from the Regression model and Smoothing Splines

6. Estimating monetary value of player based on performance

6.1. Stepwise vs Ridge

The first step towards estimating the monetary value of the player would be to build a predictive model based on player attributes and here a comparison between the accuracy of stepwise regression and ridge regression is also performed.

Using an upper bound of all available player attributes and progressing to a lower bound of none of them, the stepwise regression eliminates the unnecessary player attributes and gives a final optimized regression equation as follows:

```
  lm(formula = Value ~ Crossing + Finishing + ShortPassing + Volleys
+ Dribbling + FKAccuracy + BallControl + Acceleration + SprintSpeed
+ Agility + Reactions + Balance + ShotPower + LongShots + Aggression
+ Interceptions + Positioning + Vision + Composure + Marking +
StandingTackle + SlidingTackle + GKDiving + GKKicking, data = train)
```

Ridge regression is generally employed to reduce the complexity of the model by eliminating the no. of predictors, not by forcing them to be zero but by introducing a penalty term or bias which leads to them having a very small value, such that the sum of squared residuals is minimized towards zero.

Both of these models are trained on the training data set and after predicting the values for the testing data set, the accuracy measures of the models are as follows (R-squared(X1), Root mean squared error(RMSE) and Mean absolute error(MAE)):

```
                 X1      rmse       mae
Stepwise 0.08767151  4.909135  2.428978
Ridge    0.04656656 54.005734 53.477217
```

As one can see from the results, stepwise regression has a R-squared closer to 1, which indicates a better fit and a very small root mean squared error. Hence this model will be used further for making predictions. The complexity of the model is calculated similarly as explained in section 5.1 and the results are as follows:

```
> score_model_initial <- pred_r_squared(t_model)
> initial_overfit <- abs(summary(t_model)$adj.r.squared - score_model_initi
al)
> initial_overfit
[1] 0.002948432
>
> score_model_final <- pred_r_squared(step_model)
> final_overfit <- abs(summary(step_model)$adj.r.squared - score_model_fina
l)
> final_overfit
[1] 0.001989796
```

One such prediction is shown in figure 10 which compares the predicted worth against the actual worth of the top 5 players in figure 2 in million euros. The predicted worth for the top 5 players are lower because the historical scores are utilized, not the optimized ones from PSO.

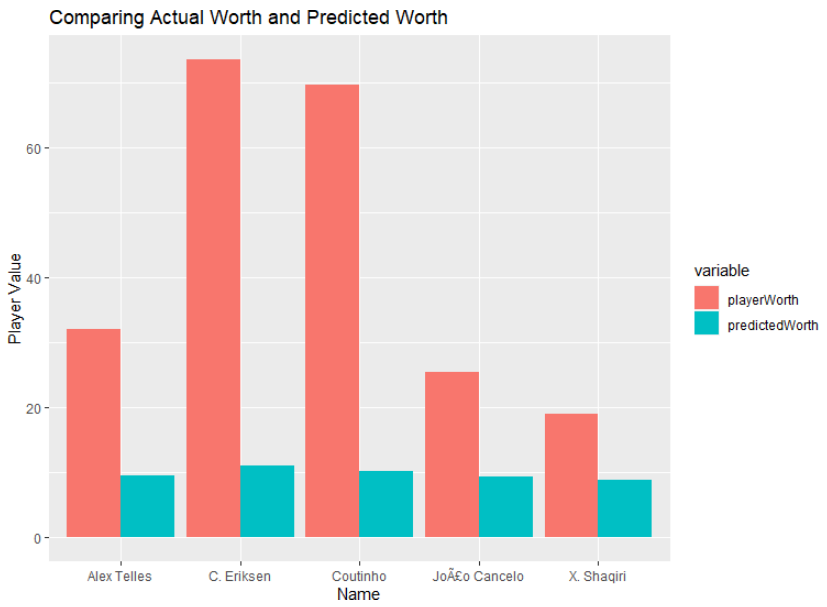

Figure 10. Actual vs Predicted Worth for top 5 players in "Crossing" skill

6.2. Predicting the value of the two players after match simulation

In Section 3.1, it was shown that the optimal left striker score depends on certain skills of the players. The regression model obtained in section 6.1 shows that the player worth depends on these skills. The basic idea here is to obtain the corresponding skill value for the optimum value from the PSO algorithm i.e. inverting the results of the score regression model in section 4.1 using 'investr' [8] and with these values estimating the player worth using the stepwise regression model in section 6.1 for the two random players selected in section 4.2.

```
pred_values_player2
## 8.191562

pred_values_player1
## 8.933734
```

Now, from these predicted worth, a comparison can be plotted with the actual worth of the two players in section 4.2 as shown in figure 11. It is seen that both players with a lower worth get a higher predicted worth from the model in section 6.1. This is due to PSO optimization which maximizes the scores for lower values.

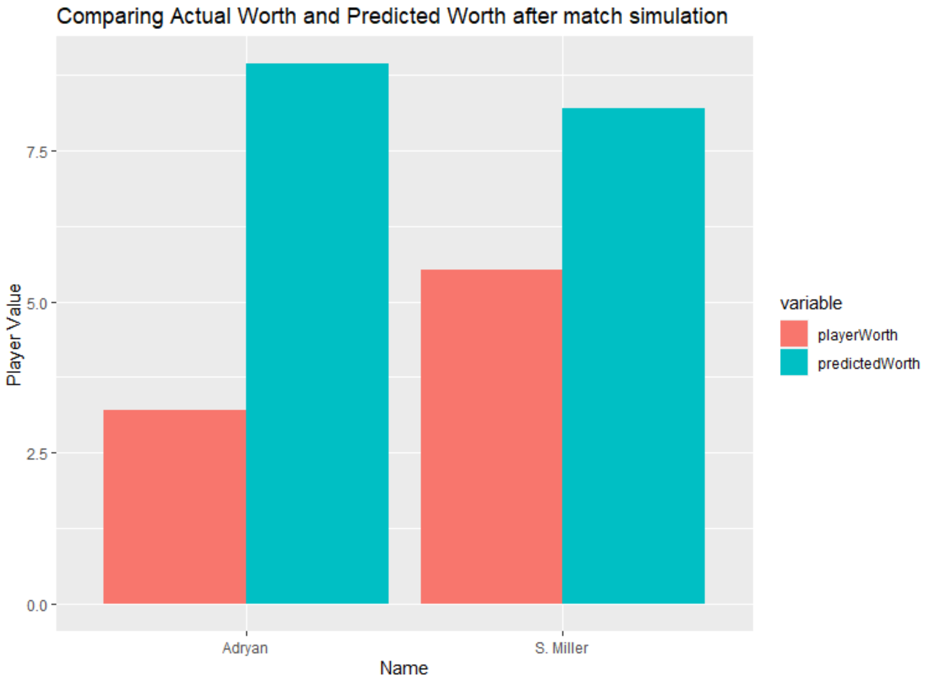

Figure 11. Comparing actual and predicted worth of two players from their optimized scores

7. Conclusion

The PSO algorithm used for the match simulation generally gives a better value than the historical data, because the conditions on the search space are constrained providing little room for decreasing values, combined with the type of optimization that has been aimed for (in this case, maximization). However, even better values are obtainable, if these values are replaced with the old values and the process is continued further.

Stepwise regression is quite an effective analysis for reducing the complexity of the prediction model and this is seen to be true in both cases – for estimating player position score and player worth, as it gets rid of the predictors that do not significantly contribute to the estimation process. Spline smoothing is just as effective with the added advantage of even lower residual errors.

Practically, it can help in putting more training for players not achieving the benchmark score, to improve their scores and achieve a higher selection possibility.

8. Future Work

- The correlation measures between the various player positions and attributes show that some attributes will never be utilized. This can be avoided by using ridge regression in building the score model
- The PSO algorithm can be updated to optimize multiple player position scores at once, to simulate a match for the entire dataset rather than only a subset of players
- The predicted worth of the player can be used to interpolate a function that outputs the total sale of tickets for a particular match

References

[1] Gadiya K, *Fifa 19 complete player dataset*, “https://www.kaggle.com/karangadiya/fifa19”

[2] Pariath R, Shah S, Surve A and Mittal J 2018, Player Performance Prediction in Football Game, *Proc. 2nd Int. Conf. on Electronics Communication and Aerospace Technology (Coimbatore)* (IEEE) pp 1148-1153

[3] He M, Cachucho R and Knobbe A 2015, Football player’s performance and market value, *Proc. 2nd workshop of sports analytics European Conf. on Machine Learning and Principles and*

Practice of Knowledge Discovery of Databases (Porto) (MLSA@PKDD/ECML) pp 87-95

[4] Chazan-Pantzalis V and Tjortjis C 2020, Sports Analytics for Football League Table and Player Performance Prediction, *Proc. 11th Int. Conf. on Information, Intelligence, Systems and Applications (Piraeus)* (IEEE) pp 1-8

[5] Perroni F *et al* 2020, Can Haematological and Hormonal Biomarkers Predict Fitness Parameters in Youth Soccer Players? A Pilot Study, *International Journal of Environmental Research and Public Health* **17** L6294

[6] Septem Riza L, *PSO,* "https://www.rdocumentation.org/packages/metaheuristicOpt/versions/2.0.0/topics/PSO"

[7] Pareto A, *Predictive R-squared according to Tom Hopper,* "https://rpubs.com/RatherBit/102428"

[8] Greenwell B, *investr,* "https://www.rdocumentation.org/packages/investr/versions/1.4.0/topics/investr"