SPPOT: A Five-Factor Team Evaluation System

Xin Zeng, Liangyu Chen, Maokang Dou, Xianmeng Wang

March 28, 2021

Abstract

Building an effective team evaluation system is of key importance in football matches, which helps a team decide the best combination of players and tactics. In this paper, we propose a simple yet strong evaluation system SPPOT that quantifies a team's performance based on the aggregated individual player attributes of the top 6 players. Our system use 5 index which represents 5 dimensions - shots, passing, physical, overall and tackle to evaluate a team.

To build the evaluation system, we firstly conduct initial feature engineering using both linear and non-linear method. Then we compare different cluster algorithms and leverage LDA to reduce dimension to get 5 index. We finally validate that our evaluation system is superior than the original team attributes in predicting the result. We effectively reduce the number of factors while enhancing predictive power.

1 Motivation and Intuition

For multiplayer sports assessment to be comprehensive, it must include variables from all players. An effective team evaluation system is important while difficult in practice. On the one hand, soccer is a team sport involving two teams, with 11 players on each side. On the other side, a few top players produce most of the observed excellent performances in statistical analysis. Coaches and team managers are interested in player analytic tools to support tactical analysis and monitor the quality of their players during individual matches or entire seasons. Not least, the soccer industry is more and more focusing on the data-driven analysis of individual player analysis. Player ranking technique has been developed to predict the performance of players and strategize formation on the team. [3] The assumption of player ranking is that a team's performance can be measured via the top players in each position. Inspired by this technique, we propose an evaluation system that quantifies a team's performance based on the top players on each individual player attribute.

We attempt to analyse a player from various performance metrics, and further aggregate player attributes into a team evaluation. In past soccer analysis, performance metrics can be classified into 2 categories: offensive and defensive. [1] While this classification is able to simulate a player's action, it lacks overall information or attributes that are related to both scenarios. Therefore we take consideration of overall performance attributes, like reactions and ball control, that could contribute to either offensive or defensive action in a match.

There are mainly two categories of our features, one relates to the overall team attributes, like *buildUpPlaySpeed* and *chanceCreationCrossing*, another one relates to each of the players in the team, like overall_rating and finishing. Intuitively, we regard the team attributes as a baseline representation of the performance, which is static. Since each team would change the players and even tactics in different matches, we regard the player attributes as a dynamic attribute which decide the different performance of the same team when encountering a similar opponent. This motivates us to conduct feature engineering from the two aspects, team attributes and player attributes.

2 Initial Feature Engineering

We first conduct data preprocessing. We use the match table as the index of our data set, joining player attributes and team attributes for both home and away teams by the nearest date. In this way, we combine both players features and team features in a single DataFrame. The columns containing large proportion of missing or meaningless values were dropped, e.g. *buildUpPlayDribbling* in team attributes, *attacking_work_rate* and *defensive_work_rate in player* attributes. And categorical variables are encoded using one-hot technique.

To better identify the most important features and modify the features to extract useful information, we implement multiple methods. We define two kinds of y, one is whether the team wins the match, another one is the difference between the number of goals scored. To capture the linear relationship between our features and y, we run Linear Regression for every single variable and compare the R^2. To capture the possible non-linear relationship between our features and y, we run GBDT model using LightGBM framework for all variables. And we use SHAP value to get the ranking of feature importance. SHAP values interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value. [2]

We first treat each of the player attributes as a single variable. For instance, for potential, we have the attribute for each of the 22 players, like *home_player_1_potential*, *home_player_11_potential*, *away_player_11_potential* and so on. By implementing both linear and non-linear methods, we find that the team attributes influence slightly to y and the top attributes are all those relating to players. After further analysis, we observe that some of the team attributes are categorical. And they have relatively more missing values than those relating to players. And for the numerical team attributes, their distributions are not distinct across different teams, which means they potentially have weak predictive power. Motivated by these observations, we focus on the player attributes in further analysis.

Considering that treating each of the player attributes as a single variable would result in lots of variables with similar information, we first aggregate the feature by their average in either home team or away team. And we run GBDT model to compute their feature importance, which gives us a general idea of the top features.

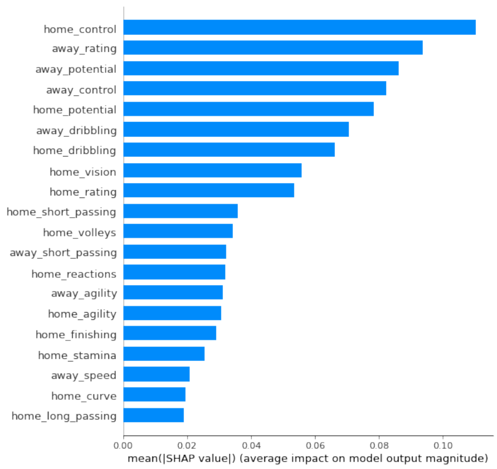

Figure 1: Feature Importance Plot Using Aggregated Features

Besides, noticing the great effect of player attributes on the final outcome of a match, an interesting question comes to our mind - would the top player in a match works as a decisive factor influencing the result of the game? We first investigate this question by looking at some "star players", like Lionel Messi and Cristiano Ronaldo. We select the matches they have participated in and see the distribution of the goals their team scored compared to the components. We see that the average difference between the number of goals their teams scored and the opponents scored are relatively high, around 2 on average. This is an interesting finding. On the one hand, if there exist some "star players" in the team, we could give them positive prediction and potentially results in an accurate result. On the other hand, for a general team, we might not have these "star player". However, there might exist some general optimal number of top players that decide the final outcome.

We further investigate how the number of top players for different features influence the y values we choose. We firstly choose 7 features which have large feature importance. For k ranging from 1 to 11, we compute the predictive variables as the difference between the average of the top k values of home team and away team. And we compute the y values as the difference of goals between home team and away team. We then run Linear Regression between y and each of them, and observe how R^2 changes with different number of top players across different features. We visualize our result using the following plot. We interestingly find that there exist some uniform patterns across different features. For most of the features, when the number of top players we choose reaches 6, we generally reach the maximum value of R^2. Then, if we continually increase k, we result in either decreasing value of R^2 or similar R^2. So, we finally conclude that $k = 6$ is the optimal number of top players in general.

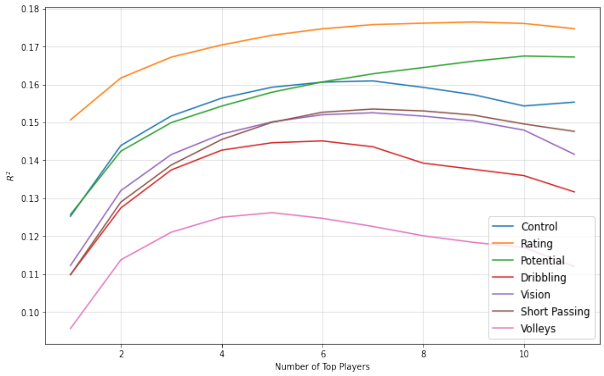

Figure 2: Relationship Between R^2 and Number of Top Players for Different Features

An additional note is how we choose the top k values. One way is first identifying the top k players through some attributes and then only using their data to represent the whole team. Another way is identifying different top k players for different attributes. We choose the second way based on data analysis and intuition. Intuitively, those with good defensive ability might not be good at shot_power. In reality, the goalkeeper would not influence how many goals scored for their team. When we represent the defensive ability of the whole team, we hope to choose the top k players for the attribute of defensive ability. Our data analysis also validates this idea. The second way could lead to a better R^2 overall compared to the first method.

3 Feature Clustering and Dimension Reduction

We select 20 player features based on the feature importance ranking, respectively as follows: overall_rating, potential, ball_control, dribbling, agility, sprint_speed, short_passing, long_passing, vision, crossing, volleys, curve, finishing, standing_tackle, sliding_tackle, marking, interceptions, stamina, positioning and reactions.

What we are going to do next is to reduce the number of factors to build a simple yet strong evaluation system. More specifically, we firstly use the clustering method to build n clusters (in this project we set $n = 5$), and then we implement a dimension reduction machine learning algorithm to reduce each cluster to one dimension. With clustering and dimension reduction, we get 5 factors evaluating the quality of one team. Note that we will also give an explanation of the meaning of each factor.

3.1 Clustering

We apply the clustering method based on the correlation matrix between 20 features. There is a heatmap showing the correlation matrix.

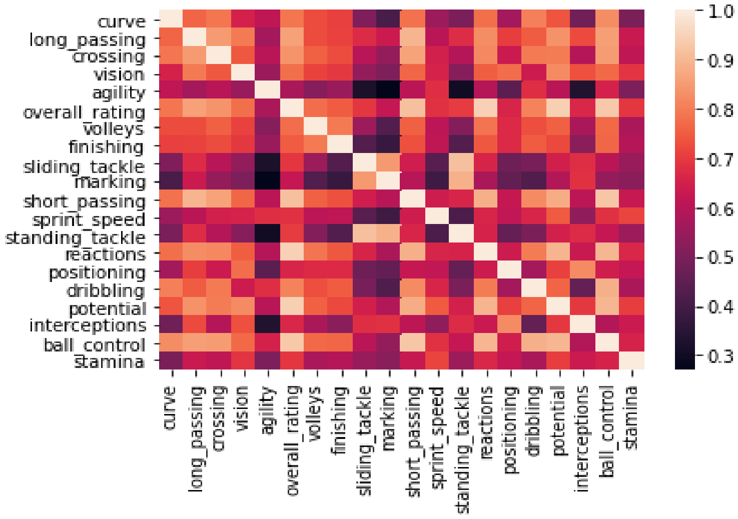

Figure 3: The heatmap of the correlation matrix

3.1.1 The Problem of Conventional Clustering Algorithms

At first we would like to utilize conventional clustering machine learning algorithms to form 5 clusters. We applied DBSCAN and flat clustering. Here we show the results:

Flat Clustering:
Cluster 0: marking, sliding_tackle, standing_tackle
Cluster 1: sprint_speed, stamina
Cluster 2: interceptions, positioning, vision
Cluster 3: ball_control, crossing, curve, dribbling, finishing, long_passing, overall_rating, potential, reactions, short_passing, volleys
Cluster 4: agility

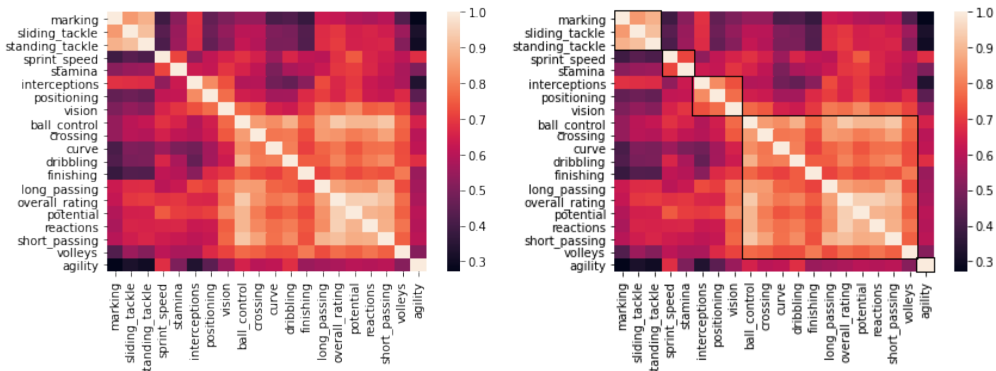

Figure 4: (left) the reordered correlation, (right) the black boxes are the clusters

DBSCAN:
Cluster 0: ball_control, crossing, curve, dribbling, finishing, long_passing, overall_rating, positioning, potential, reactions, short_passing, vision, volleys
Cluster 1: marking, sliding_tackle, standing_tackle
Cluster 2: interceptions
Cluster 3: agility
Cluster 4: sprint_speed, stamina

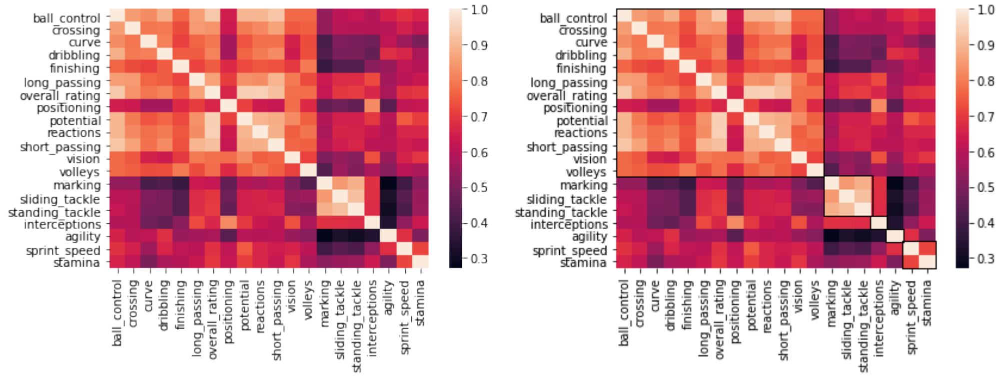

Figure 5: (left) the reordered correlation, (right) the black boxes are the clusters

We can see that the clusters from conventional cluster methods are unevenly distributed. This is a problem for two reasons. Firstly, the the evaluation system would not be that reliable because if one dimension only contains one feature, the variance could be high. Secondly, there is little explanation for the clusters with so many features. In other words, it is hard to explain them.

Why this could happen? If we take a look at the features in the selected feature list, we will notice that there are some general features (like overall_rating, potentials) which is so general that it could be highly correlated with most other features. Then, the clustering would be like that there is a big cluster with 'general' features at the center of clustering and other clusters are really small.

How to solve this problem? Simply setting the min_samples of DBSCAN wouldn't help. This method will only increase the noise attributes, here is an example.

DBSCAN:
Cluster 0: ball_control, crossing, curve, dribbling, finishing, long_passing, overall_rating, positioning, potential, reactions, short_passing, vision, volleys
Cluster 1: marking, sliding_tackle, standing_tackle
NOISE: interceptions, agility, sprint_speed, stamina

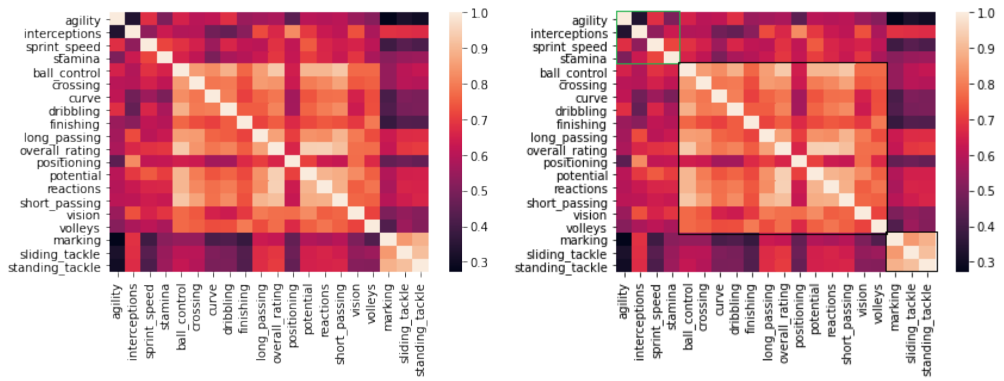

Figure 6: (left) the reordered correlation, (right) the black boxes are the clusters

The results above motivate us to build a new clustering method.

3.1.2 Block Modeling Loss

To solve the problem that the clusters are not evenly distributed, one has to find a new way to implement the clustering. Gaining intuition from Block Modeling and genetic algorithm, we design a new clustering method.

First of all, a loss function should be defined. We define the loss of the clustering is:

$$f(C_1, C_2, \cdots, C_n) = \sum_i (\sum_{p,q \in C_i} d(p,q) - \sum_{p \in C_i, k \notin C_i} d(p,k)) \tag{1}$$

Here, $C_1, C_2, \cdots, C_n$ are a set of clusters containing all elements in this cluster. $d(i,j)$ is the correlation value between i and j.

The target of this loss function is similar to other conventional clustering algorithm: that is to maximize the correlation sum inside the cluster while minimize the correlation sum between different clusters. Apart from the loss above, we add penalty to the loss equation if the cluster is not evenly distributed. The penalty equation is:

$$\textbf{penalty} = \sum_i |C_i|^2 \tag{2}$$

So the overall loss is:

$$f(C_1, C_2, \cdots, C_n) = \sum_i (\sum_{p,q \in C_i} d(p,q) - \sum_{p \in C_i, k \notin C_i} d(p,k)) - \lambda \sum_i |C_i|^2 \tag{3}$$

where λ is the hyperparameter. Later we will give some recommendations about its setting. Note that the target is to maximize this function instead of mimimizing it.

3.1.3 Genetic Algorithm

How to find the optimal clustering to maximize the loss function? Here we apply a genetic algorithm to solve this problem. Our algorithm is slightly different from the traditional genetic algorithm. The detail is below:

Algorithm 1 Genetic Algorithm for clustering

Input: Correlation Matrix(DataFrame), lambda, iter1, iter2
Output: Cluster (A dictionary showing which attribute belongs to which cluster)

n_cluster ← 5 // the amount of clusters
n_variable ← 20 // 20 variables
cluster_size ← *n_variables* / *n_clusters*
columns_num_dict ← Random_Dictionary_Initialization() // the dict for output
currentDict ← *columns_num_dict*
currentScore ← *score*(*columns_num_dict*) // score here means loss
for i in range(iter1) **do**
    optionDict ← Mutation(*currentDict*) // which means randomly change the cluster number of one attribute
    optionScore ← *score*(*optionDict*)
    for j in range(iter2) **do**
        Find the best one step crossover (which means swapping two cluster number of two attributes) from *currentDict* to get the *bestDict* and *bestScore*
        if $bestScore > CurrentScore$ **then**
            $CurrentScore \leftarrow bestScore$
            $CurrentDict \leftarrow bestDict$
        else
            break
        end if
    end for
end for
return *currentDict*

3.1.4 Math intuition behind the clustering strategy

Here we use a simple mathematical proof to show how this loss function could advocate the clustering to be more evenly distributed.

Let's consider a simple case. Except for the diagonal elements, the value of all elements in the correlation matrix is p. So for cluster $C_1, C_2, C_3, \cdots, C_n$, the loss is:

$$\textbf{loss} = \sum_i ((1-p)|C_i| + p|C_i|^2 - p|C_i|(m - |C_i|) - \lambda|C_i|^2) \tag{4}$$

where $m = \sum_i |C_i|$ is a constant (For example, here is 20). After calculation, we get

$$\textbf{loss} = (1-p)m - pm^2 + (2p - \lambda)\sum_i |C_i|^2 \tag{5}$$

we can see that $(1-p)m - pm^2$ is a constant, so to maximize this equation, there are three situations:

1. when $\lambda = 2p$, the loss is a constant.

2. when $\lambda > 2p$, from cauchy-schwarz inequality, we can get

$$\sum_i |C_i|^2 \sum_i 1 \geq (\sum_i |C_i|)^2 = m^2 \tag{6}$$

so $\sum_i |C_i|^2 \geq \frac{m^2}{n}$. This equation holds only when $|C_1| = |C_2| = |C_3| = \cdots = |C_n|$, in other words, evenly distributed. So when $\lambda > 2p$, this clustering algorithm can encourage evenly distributed clusters.

3. when $\lambda < 2p$, since $|C_i| \geq 0$, we get

$$\sum_i |C_i|^2 \leq (\sum_i |C_i|)^2 = m^2 \tag{7}$$

the equation only holds when one $|C_i|$ is m and others are 0. So when $\lambda < 2p$, the effect could reverse.

From the mathematical findings above, we know that λ should be larger than $2p$ in order to get evenly distributed clusters. As in the project, $p < 1$, so we set $\lambda = 2$ to ensure this effect.

3.1.5 Result

Each of the five clusters includes 4 features, respectively as follows:

Cluster 0: vision, finishing, volleys, positioning
Cluster 1: long_passing, curve, short_passing, crossing
Cluster 2: dribbling, agility, sprint_speed, stamina
Cluster 3: overall_rating, ball_control, potential, reactions
Cluster 4: marking, sliding_tackle, standing_tackle, interceptions

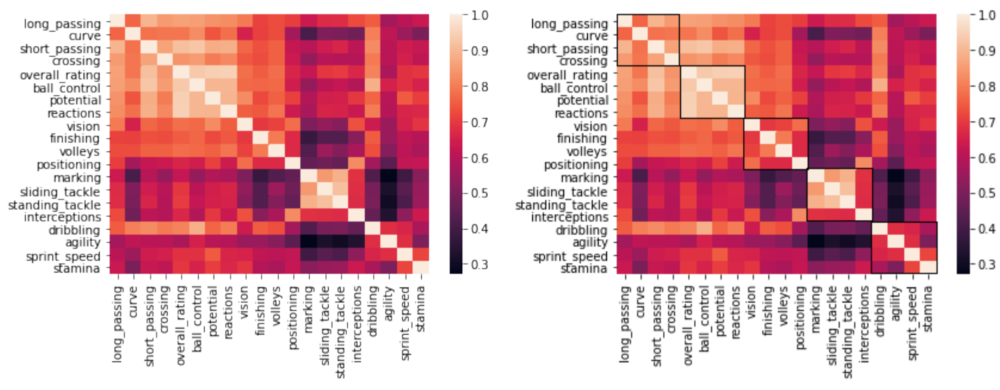

Figure 7: (left) the reordered correlation, (right) the black boxes are the clusters

Now we give some explanations to the clusters above.

1. **S**hots Cluster: vision, finishing, volleys, positioning. It is apparent that finishing is about shots and volleys is a way to shoot. Why vision and positioning are important in getting goals? Although according to the schema visions are highly-correlated with long-passing, good vision and positioning can help with achieving goals. Let's consider about the goals after the corner kick as an example. With brilliant vision and positioning, there is higher possibility in getting goals.

2. **P**assing Cluster: long_passing, curve, short_passing, crossing. Long_passing, short_passing, crossing are all about passing. Curve can be a skill in passing.

3. **P**hysical Cluster: dribbling, agility, sprint_speed, stamina. Agility, sprint_speed, stamina are about physical quality. We believe that the reason dribbling is here is because perfect physical quality can lead to better dribbling ability. Like, if you are tired or cannot speed up, dribbling can fail.

4. **O**verall Cluster: overall_rating, ball_control, potential, reactions. Overall rating and potential are a kind of general evaluation. Reactions and ball controlling are always important whenver the situation is defending, aggressing or ready to shoot.

5. **T**ackle Cluster: marking, sliding_tackle, standing_tackle, interceptions. Apparently, these are about dispossess an opponent of the ball so we name this cluster tackling.

After 5 clusters (we call it SPPOT) are formed, we will leverage dimension reduction method to reduce the features to one dimension in each cluster.

3.2 Dimension Reduction

We apply Linear Discriminant Analysis(LDA) instead of Principal Component Analysis to reduce the dimension since we believe the supervised method could better predict the result of the game.

The labels in LDA are 'win' and 'lose'. We discard games ending in a tie. Also, the features here are the feature in the home team minus the counterpart in the away team. MinMaxScaling is also applied on these features.

Here is the result of LDA:

Table 1: LDA Results

dimension	combination detail
Passing	0.95 * long_passing + 2.02 * curve + 5.50 * short_passing + 1.06 * crossing
Overall	3.54 * overall_rating + 2.37 * ball_control + 1.27 * potential + 1.09 * reactions
Shot	3.89 * vision + 1.01 * finishing + 1.83 * volleys + 2.93 * positioning
Tackle	-0.70 * marking + 4.52 * sliding_tackle + 1.34 * standing_tackle + 5.57 * interceptions
Physical	6.50 * dribbling + 1.11 * agility + 0.34 * sprint_speed + 2.19 * stamina

So here, we get 5 index representing 5 dimension in the SPPOT system.

4 SPPOT Team Evaluation System

4.1 Our product: SPPOT Team Evaluation System

Now, for each team, we can calculate 5 factors of this team to build a evaluation system. We can use a basic radar chart to show this system. Also, we build an affine mapping of the index to let them be in range between 60 and 100.

Below is an example of evaluation system in the match between FC Barcelona and Real Valladolid. The final score is 6 to 0 among these two team. Here is the radar chart. We can observe that our system gives a valid discrimination between in all five factors among these two teams.

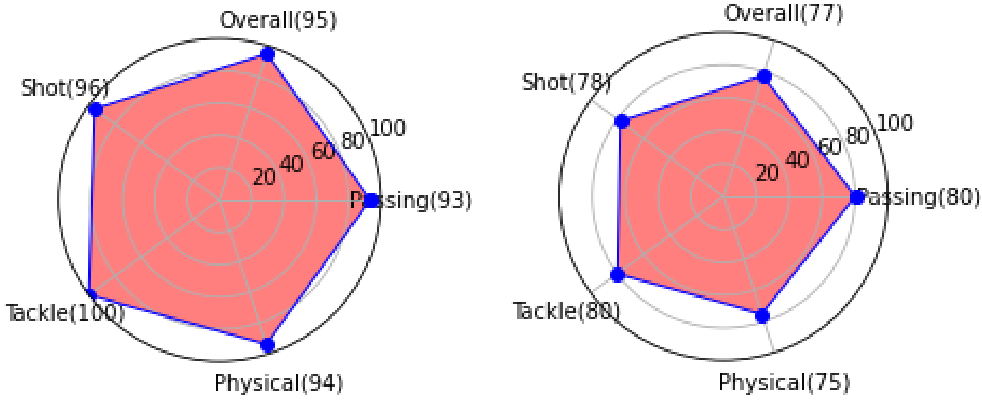

Figure 8: (left) FC Barcelona(home), (right) Real Valladolid(away)

4.2 Compared the evaluation system with original team attributes

In this subsection, we compare our evaluation system with the team attributes provided in the Citadel Datathon in order to show that our attributes outperform the original attributes.

We compare the R square between these attributes after implementing the simple linear regression. More specifically, for both original team attributes and the attributes from the evaluation system, we subtract away feature from home feature to do the prediction. The "y"(dependent variable) is the subtraction between home goals and away goals. We only selected continuous features in team attributes.

Table 2: Regression Comparison

	df	R-squared	Adjusted R-squared
Team Attributes	9	0.052	0.051
Evaluation System	6	0.181	0.181

From the table above, it is clear that the evaluation system is superior to team attributes since it has higher R-square with lower degrees of freedom. We then take a look at the detail of the regression result of SPPOT.

Table 3: Regression Results

	coef	standard error	t	$P > \|t\|$	[0.025	0.975]
const	0.4026	0.011	36.267	0.000	0.381	0.424
Passing	-0.0366	0.031	-1.184	0.236	-0.097	0.024
Overall	0.4182	0.037	11.265	0.000	0.345	0.491
Shot	0.2023	0.029	6.957	0.000	0.145	0.259
Tackle	0.0602	0.018	3.341	0.001	0.025	0.096
Physical	0.1049	0.026	4.095	0.000	0.055	0.155

All dimensions are significant except *Passing*, does that mean *Passing* should be discarded since it is not significant? Well, to answer this question, we drop the *Overall* variable to see what happens.

Table 4: Regression Results

	coef	standard error	t	$P > \|t\|$	[0.025	0.975]
const	0.4020	0.011	36.106	0.000	0.380	0.424
Passing	0.0768	0.029	2.624	0.009	0.019	0.134
Shot	0.3441	0.027	11.326	0.000	0.258	0.365
Tackle	0.1413	0.017	8.527	0.000	0.109	0.174
Physical	0.2366	0.023	10.347	0.000	0.192	0.281

We can see that *Passing* is significant in this regression. From the result, maybe because the general index contains a lot of information of passing index, the passing index seems not to be significant in the SPPOT regression. However, passing index is still useful in the evaluation system since it shows one aspect of the team.

5 Conclusion

In this paper, we firstly divide the features into two categories, player attributes and team attributes. Through initial feature engineering using both linear and non-linear methods, we mainly conclude that (1) team attributes are far less important than player attributes, (2) top players in a match could work as a decisive factor influencing the result of the game, (3) $k = 6$ is the optimal number of top players in general across different features.

Motivated by those conclusions, we aim to aggregate player attributes to build a team evaluation system. Firstly we compare multiple clustering techniques and devise a cluster algorithm to build 5 clusters. Then we leverage LDA to reduce dimension to get 5 index, representing 5 dimension in the SPPOT system. These 5 dimensions are meaningful and valid from both aspects of intuition and data analysis. Finally, we run Linear Regression to analyze the effectiveness of our evaluation system. The regression result shows that our evaluation system is superior than the original team attributes in predicting the result.

Bibliography

[1] Sixto González-Víllora, Jaime Serra-Olivares, Juan Carlos Pastor-Vicedo, and Israel Teoldo Da Costa. 2015. Review of the tactical evaluation tools for youth players, assessing the tactics in team sports: football. *Springerplus* 4, 1 (2015), 1–17.

[2] Scott Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2019. Explainable AI for Trees: From Local Explanations to Global Understanding. (05 2019).

[3] Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. 2019. PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 5 (2019), 1–27.