Technical University of Denmark

02807 Computational Tools for Data Science

What performance statistic makes a football player "good"?

AUTHORS:
Yassine TURKI, s231735
Mathilde Cros, s231738
Dimitrije Zdrale s231734
Tu Nguyen s231814

November 28, 2023

Contents

1 Introduction

Everyone knows about the importance of the football market, about the sky-rocketing wages of football players and their acquisition prices by different clubs, but is this price always justified? What part of it is purely club brand marketing? How important are the actual skills of the player in its valuation, and which skill in particular should we look at first when thinking of investing in a player?
After looking into multiple articles on this topic, we noticed that we couldn't obtain a specific formula on how the player's value on the widest skill spectrum possible was estimated nor precise explanations about a football players scout's difference in investment between two players.

We hence want to have a more precise and accurate evaluator of a football player's skill value. By using various data science and machine learning methods, we aim to identify the performance metric of football players we should look at first when evaluating their skills. Here, we are interested in the strictly technical and physical skills of players, rather than the strategic aspect of investment in players, to see whether a single feature will come out to be the most important in the player's skill value.
By doing so, this project aims to be a technical supplement to the strategic and business-related decision-making process of player recruiters. Our results could also be relevant for football clubs to objectively analyze and optimize their team's and/or player's success potential, by enhancing their decision-making processes about future investments in current and/or new players.

To get these results we decided to divide our project into two main parts.
First, after preprocessing and cleaning our dataset, we will use clustering to highlight the specificities amongst features with regards to the player's position, i.e. goalkeeper, midfielder, forward, and defender. Then, we will subdivide our dataset by each position, i.e we now have 4 datasets, and use clustering and regression methods to investigate which performance metric of players has the highest positive impact on their ranking.

2 Data

2.1 Dataset Used

To collect football players' statistics for the project, our first approach was to web scrape from well-known football statistics websites such as "Goal" or "SofaScore". However, the process was excessively time-consuming and intricate due to the dynamic nature of these websites. We hence shifted our focus towards looking for readily available datasets, where we found the website "FootyStats" (https://footystats.org/), a website with comprehensive data for football statistics from almost every league around the world. In order to understand the dataset best and avoid dissimilarities in our dataset, we decided to focus on European football only and thus downloaded the datasets of the top 7 European leagues.

2.2 Data Preprocessing and Cleaning

A lot of data cleaning and preprocessing was necessary in our dataset. Our two main issues were: handling missing data and dealing with insignificant or strongly correlated features.

After loading our 7 datasets, we first started by checking that for each league, there was the same set of columns, i.e. the same features documented for each player. We then added another column "league" to each dataset for the league that the it concerns and created one single dataset by concatenating the 7 others.

We then addressed missing data according to the feature concerned. For example, since our ground truth for our future analysis was going to be the feature "average_rating_overall", we thus dropped all players who didn't have this rating value. We also decided to drop columns of only 0 values, as they would complicate our models by requiring unnecessary extra-training. Similarly, we deleted all columns where the data was more than 80% empty. Indeed, it wouldn't be realistic to build a statistic over so few players while dismissing the rest. Finally, for the other columns that had a few missing values, we filled them using the mean of the column. Using this method allowed us to keep 5 columns that had 6 to 11 missing values only, which is very few considering we have 3479 players, so it was important not to get rid of these columns.

Secondly, we started looking for correlated features to again reduce the number of columns in our dataset. To do so, we created a correlation matrix using numpy to evaluate pairwise the relationships between the columns of our dataset. We then created a mask to identify highly correlated pairs using a correlation threshold of 0.85 and removed one column from each correlated pair.

Lastly, before normalizing the data, we dropped all columns with categorical data except from "position" and "full_name" as we will later be using these in both clustering in part 1 and to separate our dataset into sub-frames in part 2 of the project. The reason for us dropping those columns is that after further inspection of the 4 columns concerned ("birthday_GMT", "season", "Current Club" and "Nationality"), we found that either there was another column which represented the same statistic numerically ("age" for "birthday_GMT") or that the feature wasn't relevant to our study since we wanted to focus only on the strictly technical and physical features of the players ("Nationality" was thus not very relevant). For the last 2 features, we also judged that these were highly correlated with the player's skillset, and thus weren't in themselves a skill, like "Current Club" which is good if the player is already good. We then encoded the feature "position", so that we would later be able to use it more easily in our models. We thus associated and replaced "Goalkeeper" with 1 , "Defender" with 2, "Midfielder" with 3 and "Forward" with 4. We also dropped the features "rank_in_league_top_attackers", "rank_in_league_top_midfielders" and "rank_in_league_top_defenders" as they were also obviously linked to our ground truth feature "average_rating_overall" without having any information about the skillset of the player, except a result from a weighting of their performances to evaluate how good they are. We then normalize the remaining dataset of 3479 players and 141 features to have consistent data in order for our comparisons to be meaningful.

Columns dropped	**Number of Columns**
Only 0 values	31
More than 80% empty data	8
Correlated	87
Categorical Data	4
Linked to our Ground Truth Feature	3

Table 1: Resume of our Feature Reduction and the Associated Amount of Dropped Columns

Total Dropped Columns	**Total Remaining Columns**
133	141

Table 2: Resume of our Data Cleaning

3 Part 1: Clustering Analysis Into Positions

We were interested in showing that it was necessary to separate the players according to their positions before trying to determine the most important feature in their valuation as we supposed that some features were relevant only to some positions. To do so, we decided to implement a Clustering algorithm after dropping all columns that we didn't need right now or and the position of the player, i.e. "league", "player" and "full_name", in a new sub-dataset called "clustering_data_1".
We then ran a k-means clustering for different values of k and plotted the associated Within-Cluster Sum of Squares (WCSS). Afterwards, we used the elbow method on this plot to find that the optimal k for the algorithm was in fact k=4. At that point, since a player can only have 4 different positions, we supposed that this clustering corresponded to this feature. To check our hypothesis, we plotted the clustered data points', i.e. players', "average_rating_overall" feature as a function of features we supposed were specific to one position or another of a football player.

Here were our results:

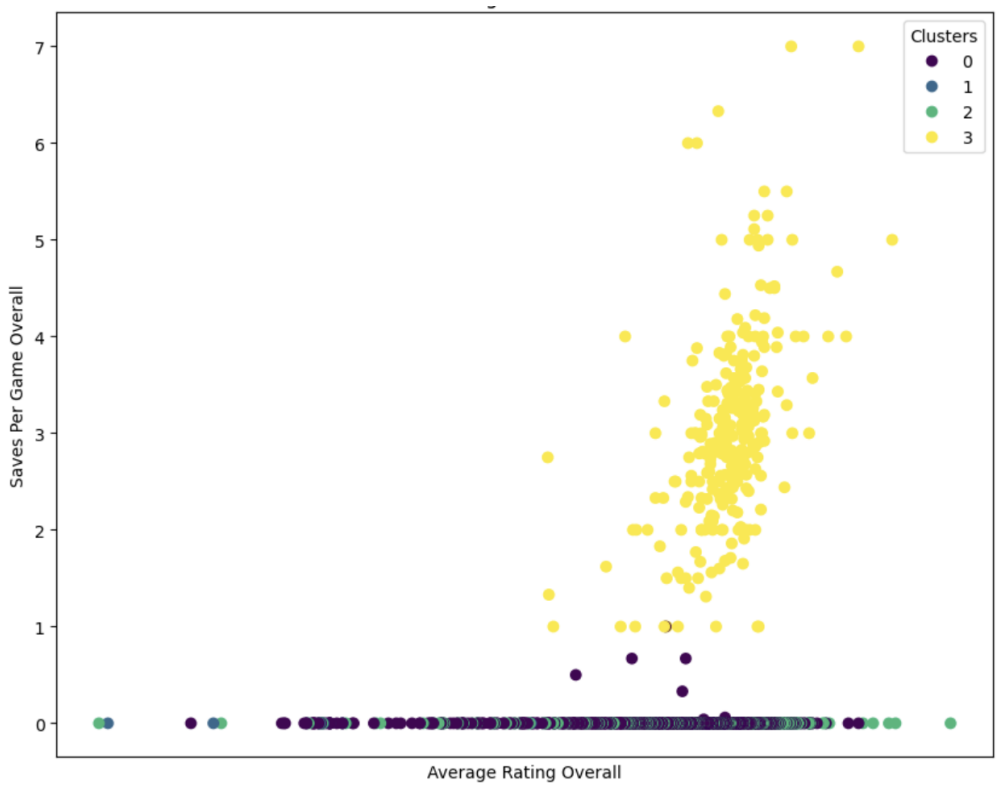

Figure 1: Plot of the Clustered Data Points per Goals Saved per Game

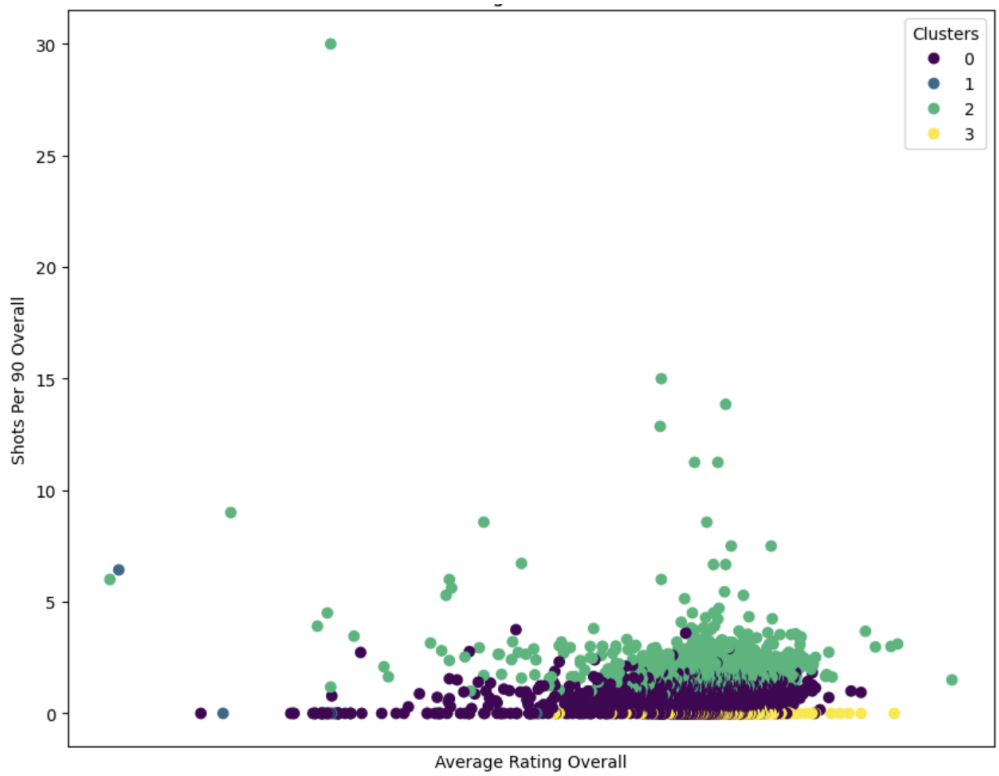

Figure 2: Plot of the Clustered Data Points per Shots Taken per 90min

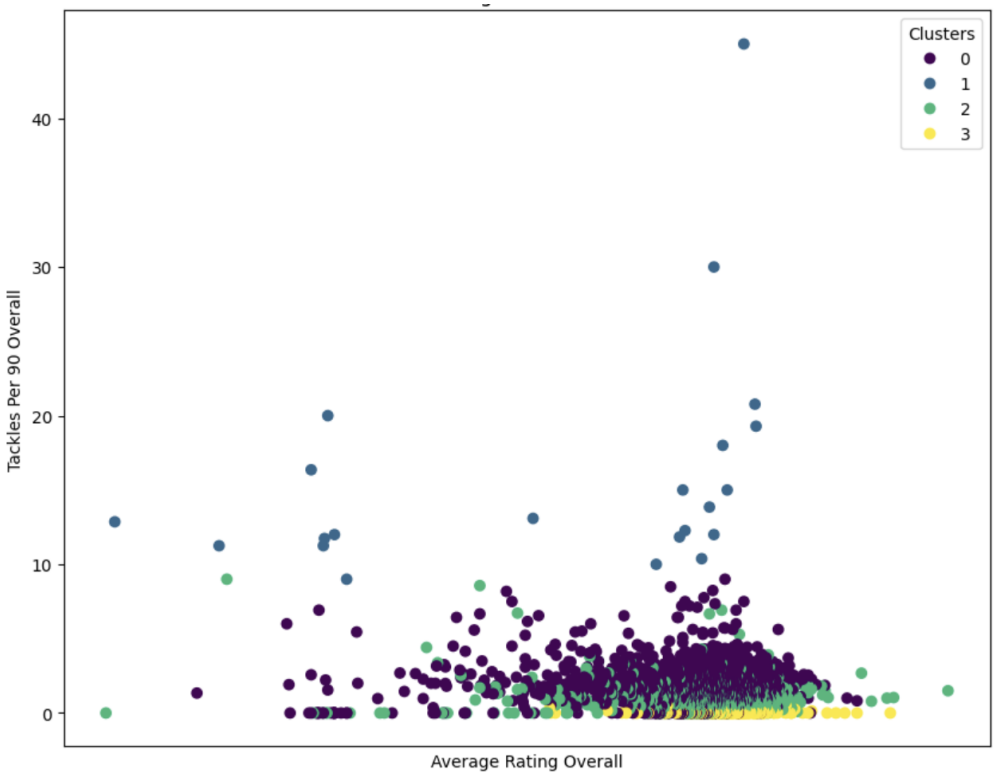

Figure 3: Plot of the Clustered Data Points per Tackles Made per 90min

Here we can see that there is indeed some clustering that is possible by position. From these results, we see that in the first plot with "saves_per_game_overall", all the players in the yellow cluster correspond to goalkeepers as they are obviously the players that will have non-zero statistics for goals saved. From the other two plots, we first could guess for the plot of "shots_per_90_overall" that the players in the green clusters are forwards as they are those who shoot most often, while in the plot of "tackles_per_90_overall" the players from the blue cluster are those who tackle most often so they are most likely the defenders. Lastly, for the players in the purple cluster, since they are the second most players to tackle and shoot, they would logically be the midfielders.

As we clearly have clusters of players sharing similar features, it is more relevant to cluster players within the same position before looking at these features in detail.

4 Part 2: Predicting the Most Valuable Feature per Position

4.1 Clustering Analysis by Position

In the second part of the project, we wanted to look at the best feature per position of players so we needed to separate our dataset into 4 sub-datasets according to the "position" feature. Using the initial dataset (called "df_cleaned") we dropped all the features "league", "position", "annual_salary_eur", "annual_salary_eur_percentile", "shirt_number" as they weren't relevant in our analysis and saved this dataset in "clustering_data_2". We then categorized the players into 4 subsets where we ended up having 279 goalkeepers, 1158 defenders, 1147 midfielders and 895 attackers.

To further tailor the clustering process, we manually selected relevant statistics for both goalkeepers and outfield players. The selected features include various performance metrics such as age, assists, goals, passing accuracy, defensive actions, and more, reflecting the diverse playing styles of different positions. We did this manually as there were many obvious redundant features from our datasets, for example, "goals_per_game_overall" and "goal_per_90_overall", which are basically the same. After doing this reduction, we had a third of the features left.

In the context of our football player analysis, we tested various clustering algorithms such as K-means, Spectral Clustering, Agglomerative Clustering, and Bisect K-means. K-means clustering is used for its simplicity and efficiency, making it a reliable starting point for our project, already since part 1. Spectral Clustering, on the other hand, is employed to capture complex patterns in the data, particularly when clusters are non-convex or exhibit intricate structures. However, it is mostly used for a small number of clusters, which could be a disadvantage in our case. Agglomerative Clustering is used thanks to the ability of hierarchical exploration of player similarities, as it successively merges similar clusters, revealing a nested structure that aligns with potential player roles or playing styles.

Initially, we attempted to identify the optimal number of clusters using the elbow method. However, the elbow point was too small (around 10 for all positions) so we would have too many players in a cluster, which was not ideal. We thus compared the performance of K-means, Agglomerative, and Spectral clustering across all player positions with the Silhouette score, Davies-Bouldin score, and Calinski score to find a better method.
As a brief reminder of these scores:

- **Silhouette Score**: A measure of how similar an object is to its own cluster compared to other clusters. Higher values (closer to 1) indicate better-defined clusters.
- **Calinski-Harabasz Score**: Also known as the Variance Ratio Criterion, it measures the ratio of between-cluster variance to within-cluster variance. Higher scores suggest better-defined clusters.
- **Davies-Bouldin Score**: Evaluates the compactness and separation between clusters. Lower scores indicate better-defined clusters.

Now while the silhouette score and the Davies-Bouldin score provided valuable insights into cluster quality, our emphasis lied in creating clusters where players within the same group shared similar characteristics. Therefore, we prioritized the Within-Cluster Sum of Squares (WCSS) loss metric when using the K-means clustering algorithm. Unlike silhouette and Davies-Bouldin scores, WCSS assesses the compactness of clusters, aligning more closely with our objective of forming internally cohesive groups.

Now, the task was to determine the optimal method and number of clusters for each player position. As WCSS served as our primary assessment criterion, aiming to minimize the loss, an inherent challenge arised. While increasing the number of clusters tended to decrease WCSS, we needed to strike a balance. To address this, we introduced silhouette score, Davies-Bouldin score, and Calinski-Harabasz score as threshold constraints. Silhouette score, for instance, reflects the extent of separation between clusters. Hence, we selected the number of clusters ensuring that silhouette score remained sufficiently high (¿= 0.1), indicating well-defined groups, while Davies-Bouldin score remained suitably low (¡= 1.9), signifying reasonable compactness between clusters. This approach allowed us to form clusters with a balance between cohesion and separation.

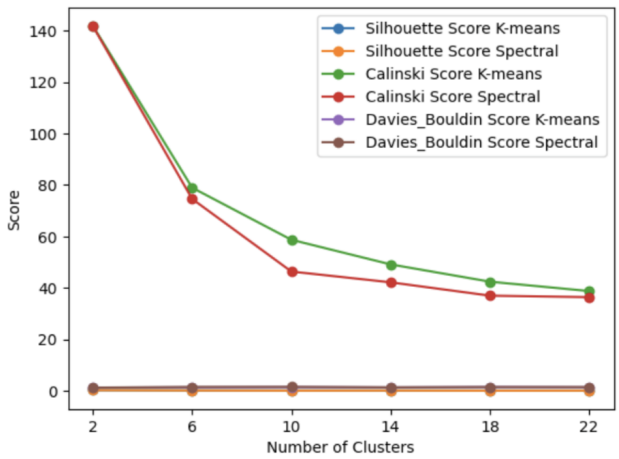

Figure 4: Assessing with 4 criteria K-means and Spectral Clustering of Goalkeepers

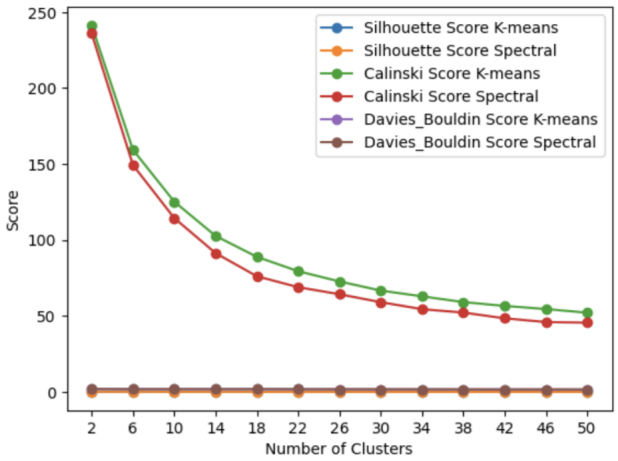

Figure 5: Assessing with 4 criteria K-means and Spectral Clustering of Defenders

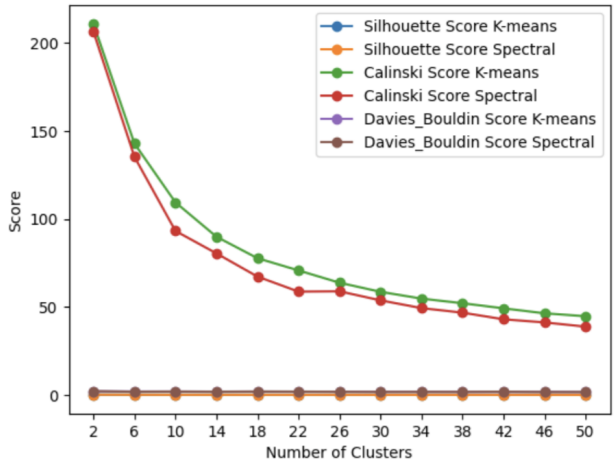

Figure 6: Assessing with 4 criteria K-means and Spectral Clustering of Midfielders

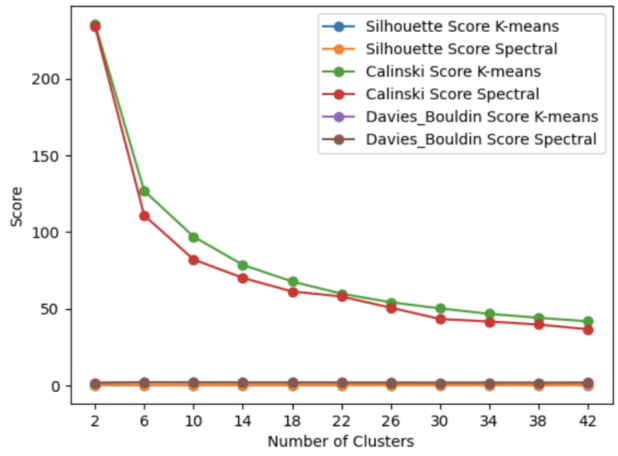

Figure 7: Assessing with 4 criteria K-means and Spectral Clustering of Forwards

After checking that K-means performed the best out of the 3 models, we wanted to include an extra method which is the Bisect K-means - a slightly different version of K-means to test if it performed better than the original K-means. We thus implemented Bisect K-means from scratch. The idea of Bisect K-means is that in each iteration, we choose the worst cluster and "bisect" it into 2 new clusters using K-means with n_cluster=2. The definition of a bad cluster can vary, however in our case, we used the sum of squares as the criteria to evaluate it. Thus, after each iteration, we removed the worst cluster and added 2 new clusters. We repeated this process until we get k clusters as desired and then plot the WCSS loss to compare it with K-means.

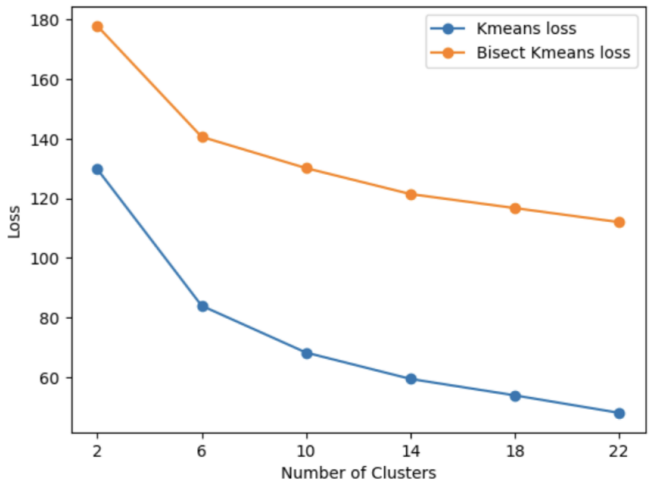

Figure 8: Comparison of K-means and Bisect K-means for Goalkeepers

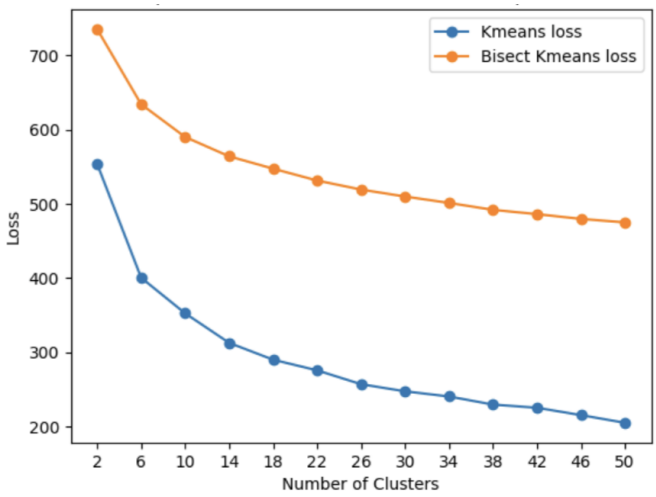

Figure 9: Comparison of K-means and Bisect K-means for Defenders

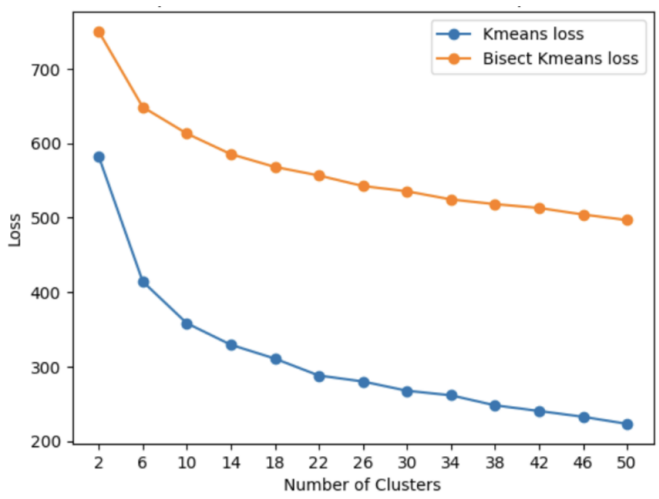

Figure 10: Comparison of K-means and Bisect K-means for Midfielders

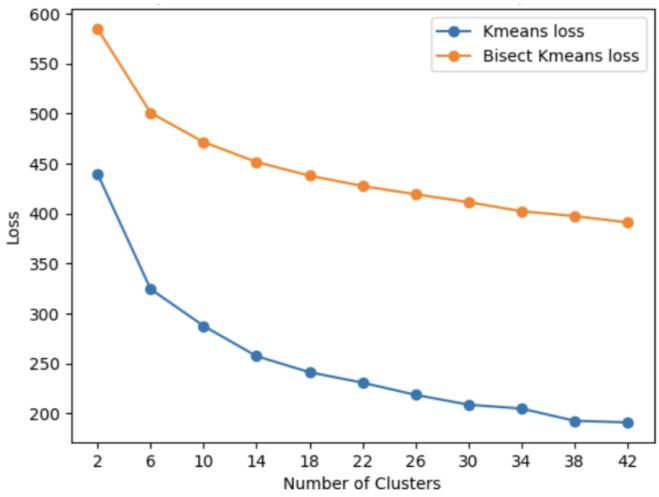

Figure 11: Comparison of K-means and Bisect K-means for Forwards

After plotting the WCSS score and printing the 3 criteria scores, we decided that K-means was the best model and that the number of clusters were 22, 50, 50, and 42 for Goalkeepers, Defenders, Midfielders, and Attackers respectively. The silhouette score was around 0.1 for all 4 positions, which indicated that the players are not in the wrong clusters, but there might be overlapping among clusters. We then ranked all players within a position and looked at the clusters to see if similarly ranked players were within the same cluster. Some of the clusters are shown below:

Cluster 26 (Sorted by Age | Mean Rating >= 7):

	full_name	age	average_rating_overall
6	Dušan Tadić	34	8.02
11	Kevin De Bruyne	32	7.76
130	Florian Kainz	31	7.27
81	James Maddison	26	7.36
30	Pedro Gonçalves	25	7.55
87	Bukayo Saka	22	7.34
738	Michael Olise	21	6.95

Cluster 24 (Sorted by Age | Mean Rating >= 7):

	full_name	age	average_rating_overall
4	Lionel Messi	36	8.15
13	Antoine Griezmann	32	7.73
15	Neymar	31	7.71
48	Oussama Tannane	29	7.45
60	Vito van Crooij	27	7.42
28	Václav Černý	26	7.58
1	Cody Gakpo	24	8.35

Cluster 38 (Sorted by Age | Mean Rating >= 7):

	full_name	age	average_rating_overall
499	Christopher Trimmel	36	7.02
17	Kieran Trippier	33	7.65
172	Cristiano Biraghi	31	7.22
163	Maximilian Wittek	28	7.23
139	Borna Sosa	25	7.25
69	Pedro Porro	24	7.39
189	Yanis Hamache	24	7.21

Cluster 0 (Sorted by Ranking | Mean Rating >= 6.8):

	full_name	age	average_rating_overall	ranking
20	Peter Leeuwenburgh	29	7.61	6
71	Daniel Bentley	30	7.38	13
348	Leo Román	23	7.10	48
573	Diego Mariño	33	7.00	83
687	Alex Meret	26	6.97	102
802	Jan Oblak	30	6.94	112
1050	Łukasz Skorupski	32	6.88	132
1386	Unai Simón	26	6.81	163
1630	Alexander Meyer	32	6.76	183
1773	Pietro Terracciano	33	6.72	195
1927	Fabian Bredlow	28	6.69	205
2279	Mile Svilar	24	6.62	220

Cluster 4 (Sorted by Ranking | Mean Rating >= 6.8):

	full_name	age	average_rating_overall	ranking
9	Asmir Begović	36	7.91	3
24	Fabian de Keijzer	23	7.58	7
46	Samuel Portugal	29	7.46	9
159	Donovan Léon	31	7.23	21
228	Frederik Rønnow	31	7.17	29
234	David Raya	28	7.17	29
357	Jan Olschowsky	21	7.09	53
427	Aitor Fernández	32	7.06	61
535	Alban Lafont	24	7.01	79
741	Sam Johnstone	30	6.95	108
960	Wladimiro Falcone	28	6.90	121
1188	Jeffrey de Lange	25	6.85	143
1232	Manuel Riemann	35	6.84	151
1530	Bingourou Kamara	27	6.78	177
2086	Paul Bernardoni	26	6.66	213
2258	Marcin Bulka	24	6.62	220
3189	Federico Marchetti	40	5.90	273

Figure 12: Example of Five Different Clusters and Different Features Printed

We can observe that although the best players are not in the same clusters, there are still proximities within players in clusters (for example, number 3, 7 and 9 are within the same cluster). Therefore, we further develop our investigation to see which features actually contribute to having a high score. For this task, we will use Machine learning.

4.2 Machine Learning Methods

Our results from the previous clustering analysis by position gave us more insight to various importance of features in a football player's ranking. However, the interpretability of the results were not conclusive enough in our opinion, hence we decided to gain deeper insight into our data to figure out what was the key feature making players stand out by using various regression models and comparing their predictions. We decided to train them to minimize the Mean Squared Loss (MSE) and Mean Absolute Loss (MAE) and then used the absolute value of regression coefficients for our models to evaluate feature importance.

We first started using linear regressions. Starting off with this machine learning model was suitable as we supposed that the relationship in our data is linear, and as the name tells us, linear regressions are particularly well suited to fit a line that will best represent this relationship. The Mean Squared Error here was acceptable, however, linear regressions are very sensitive to outliers and we noticed that this impacted its performance especially with regards to the goalkeepers group since it had the fewest amount of players/data.
We thus decided to add penalties to our loss function, such as the L1 norm (Lasso), the L2 norm (Ridge), and a combination of both (ElasticNet). Indeed, the error was diminished and these models would only consider the features relevant to our analysis and reduce the coefficient of non-important features. This is due to the fact that Lasso encourages sparsity by pushing some coefficients to exactly zero while Ridge prevents overly large coefficients by adding the squared values of the coefficients as a penalty term, and ElasticNet combines both penalties.

Additionally, we decided to look further than regressions by pursuing our analysis using tree models. Notably, we used a decision tree regressor and then optimized our predictions using bagging (Random Forest regressor) and boosting (XGBoost regressor). Thanks to the training of multiple decision trees, these last two methods allow us to reduce the effect of outliers, avoid overfitting our data, and get consistent predictions.
We can explain such behaviours of these models by the fact that the concept behind decision trees is that they recursively split the data based on feature values to create decision rules (nodes) which lead to different predictions (leaves). The concept of bagging and boosting are just different methods using multiple decision trees at once. Random Forest Regressors are generally more robust models as they randomly merge predictions of multiple decision trees, but take longer to train, while XGBoost Regressors, i.e. extreme gradient boosting, considers the decision trees sequentially and thus trains itself by emphasizing previous mispredictions. Here both methods were thus suitable to try out.

In the plots below, we denote the different MSE we got for each model, as well as the most important feature-determining predictions. It is relevant to note that we chose to drop a lot of redundant / highly correlated features, such as "minutes_played_overall", which obviously plays a pivotal role in predictions, as a player with lots of experience would have lots of passes, shots, duels, etc. However, this feature encompasses so many that it reduces the interpretability of our models.

Here were our results:

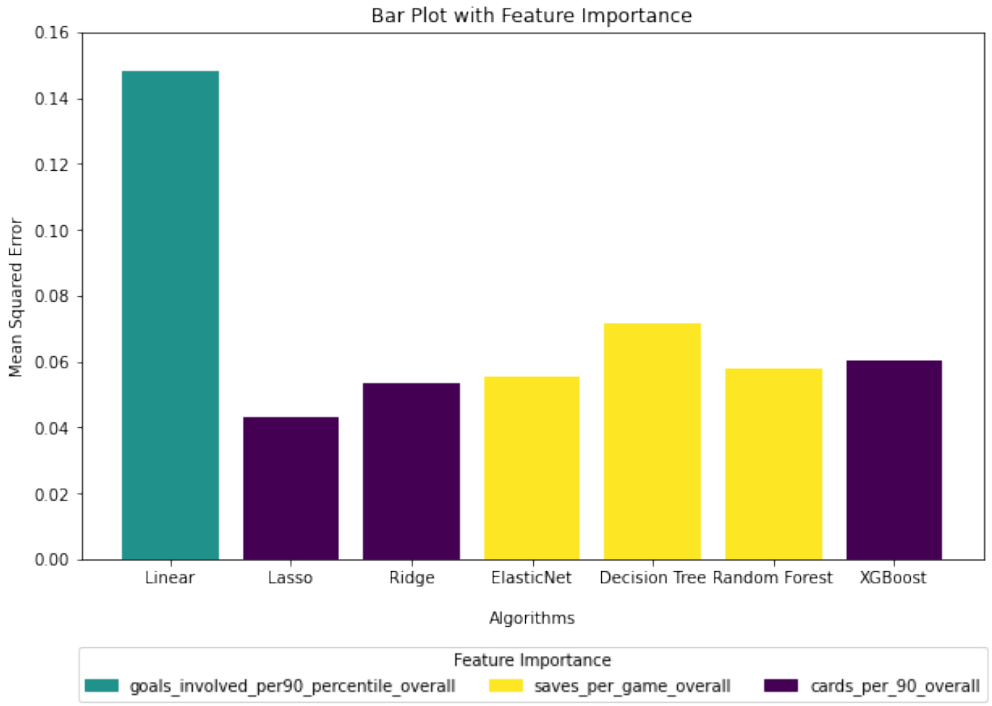

Figure 13: MSE for the Predicted Best Feature for Goalkeepers of Each of the Algorithms We Implemented

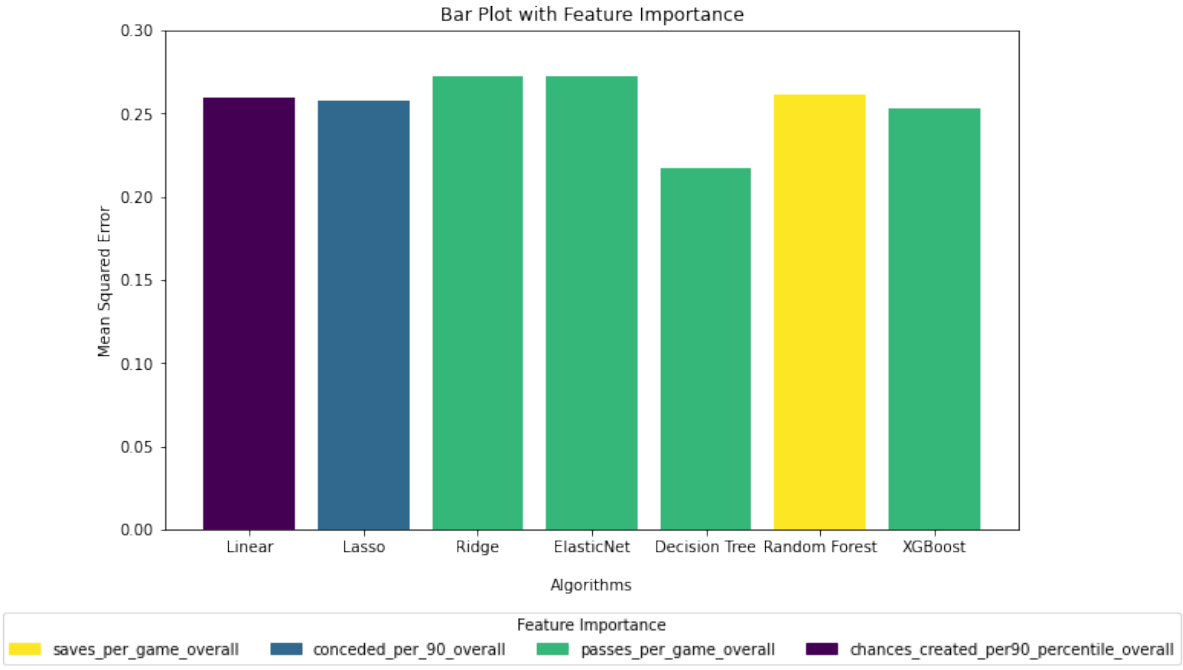

Figure 14: MSE for the Predicted Best Feature for Defensers of Each of the Algorithms We Implemented

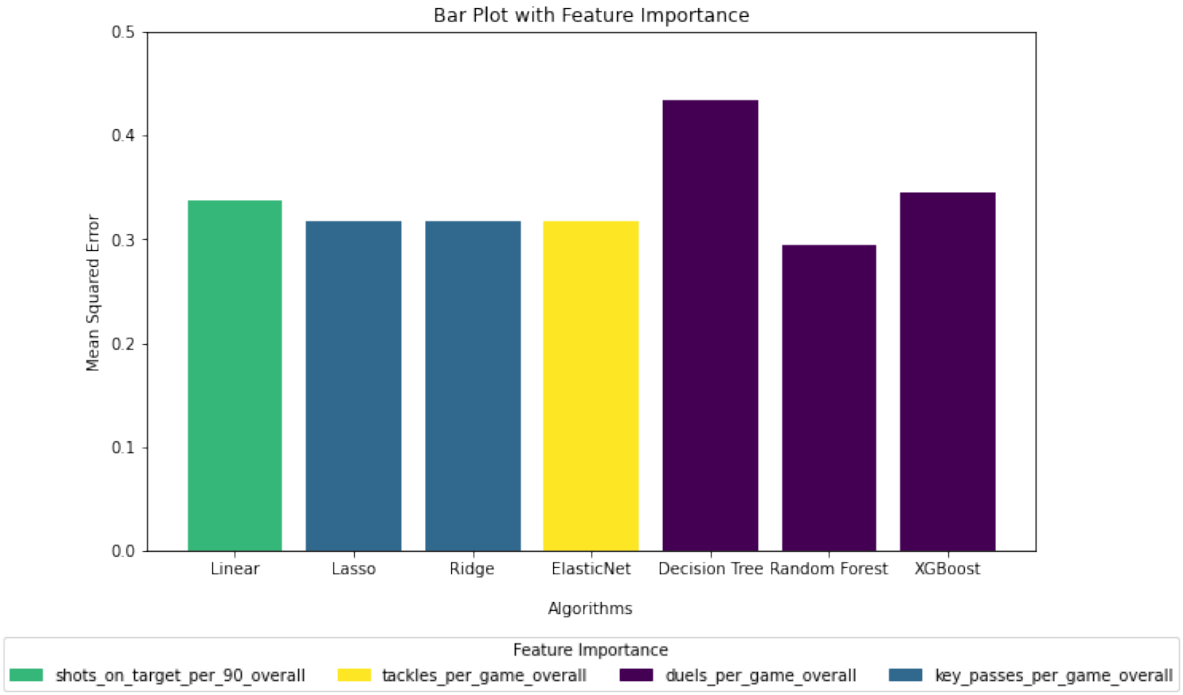

Figure 15: MSE for the Predicted Best Feature for Midfielders of Each of the Algorithms We Implemented

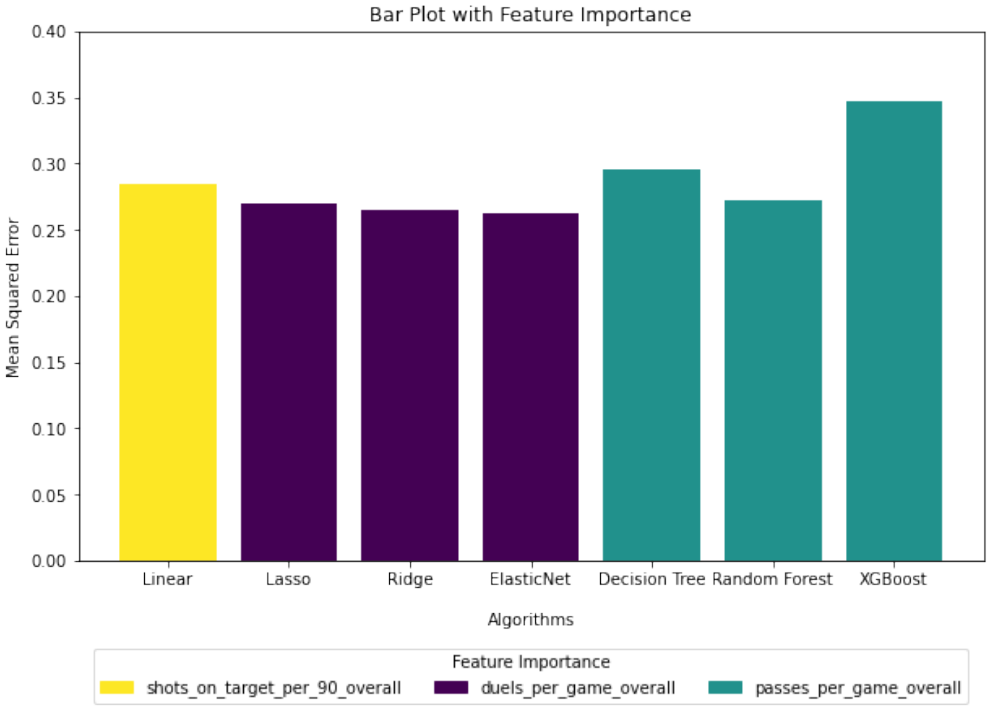

Figure 16: MSE for the Predicted Best Feature for Forwards of Each of the Algorithms We Implemented

As we can see the most important feature in being a good goalkeeper is the number of cards per 90minutes which can seem a bit surprising, but a goalkeeper getting a card also means that he is playing very actively and attempting trickier saves. When we also look at the second most important feature being goals saved per game, this also makes sense. For a defender, it is the number of passes per game and this is probably due to the fact that if a defender passes a lot, it means he managed to retrieve the ball from the opponent and pass it back to his team.

For midfielders, we can see that the number of duels is the prominent feature, as midfielders need to both secure the ball from the opponent and face the goalkeeper to score. Finally, the best forwards are the ones with a high number of duels per game and then passes as second best, which means that they know how to not loose the ball against the other team's defence line and how to make good passes in order to lead their team to a goal. Overall, we think these features make sense, especially considering that all the predictions of our models, even the third best also make sense (for example, the third most important feature for forwards is the amount of shots on target per game).

5 Results

All in all, we found the following conclusive results on the most important feature for each football player position:

Position	**Most Important Feature**	**Second Most Important Feature**
Goalkeeper	cards_per_90_overall	saves_per_game_overall
Defenders	passes_per_game_overall	conceded_per_90_overall
Midfielders	duels_per_game_overall	key_passes_per_game_overall
Forwards	duels_per_game_overall	passes_per_game_overall

Table 3: Resume of our Results

To come up with these results, we decided from our results and the plots in the section above that the best feature would be the one that is predicted by most algorithms, and then with lowest Mean Squared Error in case of a tie.

We also wondered how uncorrelated our features were even after our thorough data preprocessing. We thus decided to implement PCA with a GridSearch algorithm to determine the best number of components to optimize the accuracy of a Logistic Regression. PCA was the suitable model choice in our opinion as it summarizes a big data set into smaller indices which can be easier to visualize and/or analyze. We obtained that the best number of principal components was n=89, whereas the "optimaly reduced" dataset we used throughout the project still had 141 features. Obviously, we choose not to incorporate the PCA in our study directly due to the low interpretability of these components since they are new components, and we are trying to look at the importance of the specific ones in our dataset, i.e. without changing them, on the ranking of the players. However, this result tells us that it is possible to find some information with less features. This implies multiple correlations between the players' features, which makes our results on the prediction of the "only one best feature" open to discussion.

6 Outlook

With the increase of interest in the 2023 FIFA Women's World Cup, coupled by the increase in criticism on the skills level of these female players, we have further come to ask ourselves whether the performance metric having the highest influence in qualifying a male football player as a "good" player, is the same for female players.
As a next step to this project, this would allow our results to be used to promote female football players' skills and hence attract more attention to their competitions.

7 Appendix and Group Member Contributions

At the following Github repository: Computational-Tools-For-Data-Science-Group46, you will find the Jupyter Notebook of our project with detailed code, comments and explanations. You will also find a document detailing each group member's contribution to the project, and a README file provinding clear instructions on how to use our code and replicate our analysis.