
Soccer Prediction
Manjit Ullal | Shubhanshu Gupta | Kathan Patel | Pengyu Wang

Soccer in general is unpredictable and the element of surprise is what keeps everyone hooked to the game. However this unpredictability is not good for someone who is part of the game, for instance a Manager. Our aim, through the project is to create insights related to team and players performance which could help a Manager of the team to make guided decisions. We intend to answer the above questions.

We will use match table to answer the classification of Team's Win/Loss, and use the Player table to predict Player value and cluster players.
For classification, the Aim is to look at the Team's characteristics and try various approaches and attributes, and verify which set of attributes produce better predictions.
Then we aim to predict how valuable a player is looking at his performances and characteristics, through the use of regression.
In clustering, we aim to find group of players with similar attributes, such that they can be probable replacements for each other.

These insights are intended to help a Manager better prepare for a Match, find replacements for a player who is out of contract and find the market value of players.

Dataset

- Kaggle - European soccer database
- Scraped data - Sofifa.com

Match Table

Rows	25,979
Columns	115
Null Values	1458

country_id	league_id	season	stage	match_api_id	home_team_api_id	away_team_api_id	home_team_goal	away_team_goal
1	1	2008/2009	24	493016	9996	8635	1	1
1	1	2008/2009	26	493041	9996	8571	1	0
1	1	2008/2009	28	493053	9996	9998	3	2
1	1	2008/2009	30	493071	9996	9985	0	3
1	1	2008/2009	32	493096	9996	9993	3	1

Player Table

Rows	2,35,000
Columns	88

Year	Name	Positions	Age	Rating	Potential	Team	Contract	Height	Weight	Foot	Best Overall	Best Position
2008	Ronaldinho	CAM	27	91	93	FC Barcelona	2010	5'11"	179lbs	Right	91	CAM
2008	Cristiano Ronaldo	RW	22	91	94	Manchester United	2012	6'1"	183lbs	Right	91	CF
2008	T. Henry	ST	29	91	91	FC Barcelona	2011	6'2"	183lbs	Right	91	ST
2008	G. Buffon	GK	29	91	93	Juventus	2012	6'3"	181lbs	Right	91	GK
2008	A. Nesta	CB	31	91	92	Milan	2011	6'2"	174lbs	Right	91	CB
2008	Kaká	CAM	25	90	91	Milan	2009	6'0"	161lbs	Right	90	CAM

The dataset is of European Soccer games and is taken from Kaggle. Database consists of three major tables – Matches, Teams and Players table.

Match table - The match table includes every single European soccer match that happened between the years of 2008 to 2016 and has detailed information with a total of 115 columns. Attributes include country, season, home team, away team, all 11 players and their position with index of (X,Y) on the field for each team. However, for other information like number of defensive foul, number of yellow card given to specific player and number of cross & corner, 67% of the data have missing value, and other 33% of the data are xml format raw data with specific information like when the yellow card is given that need to parse.

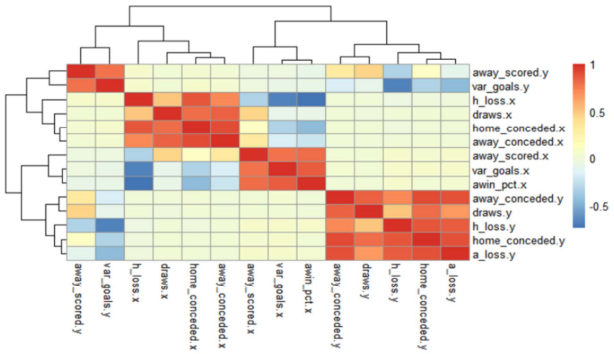
Team table - The team table has data regarding build up play, defense speed, chance creation of different teams. It has 1458 rows with no missing values.

Players table – Data was available from 2008 to 2016. We scraped the data of players from sofifa.com from 2017 to 2020. The players table includes records of top 18000 players for each season in different European leagues. Table has features such as Name, Age, Height, Weight, Preferred Foot, etc. of the player. It also has details about the player abilities and has around 35 different attributes such as Volleys, Marking, Free kick , Goalkeeper Reflexes, etc. to name a few. Apart from that, the table had details about the Team of the player, Contract Details, Value, Wage and Release Clause of the player. In totality the table had around 2,35,000 rows with several null values in different columns.

Feature Engineering

Match Table: Engineered Features

home_team	matches	h_matches	a_matches	tot_scored	home_scored	away_scored	wins	losses	draws	h_wins	a_wins	h_loss	a_loss	mean_goals	var_goals	win_pct	loss_pct	hwin_pct	awin_pct	points
Arsenal	304	152	152	573	306	267	170	61	73	97	73	21	40	1.884868421	2.003202623	0.559210526	0.200657895	0.638157895	0.480263158	583
Aston Villa	304	152	152	335	179	156	86	130	88	45	41	57	73	1.101973684	1.220590151	0.282894737	0.427631579	0.296052632	0.269736842	346
Birmingham City	76	38	38	75	38	37	21	29	26	14	7	7	22	0.986842105	0.546491228	0.276315789	0.381578947	0.368421053	0.184210526	89
Blackburn Rovers	152	76	76	175	98	77	42	71	39	29	13	26	45	1.151315789	1.082912165	0.276315789	0.467105263	0.381578947	0.171052632	165
Blackpool	38	19	19	55	30	25	10	19	9	5	5	9	10	1.447368421	1.118776671	0.263157895	0.5	0.263157895	0.263157895	39



Players Table: Engineered Features

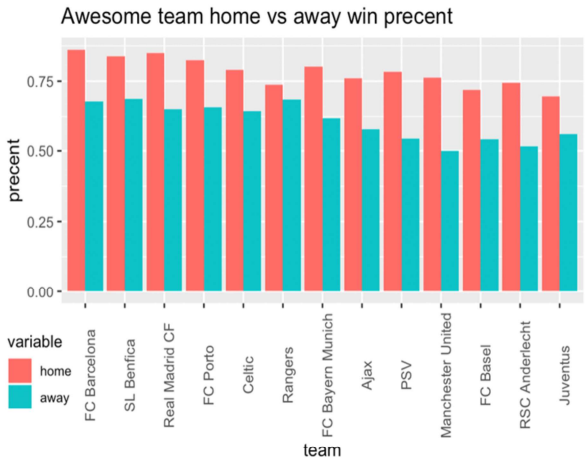
TotalPositions	Contract Remaining	Category	Goals_Per_Appearance	MinutesPlayedPerAppearance	Last_yr_val
3	2	Midfielder	0.07142857	64.46428571	23000000
3	1	Midfielder	0.6969697	82.06060606	13000000
3	2	Midfielder	0.17241379	81.89655172	235000000
4	3	Midfielder	0.24324324	82.97297297	10000000
2	3	Midfielder	1.03030303	88.87878788	37000000

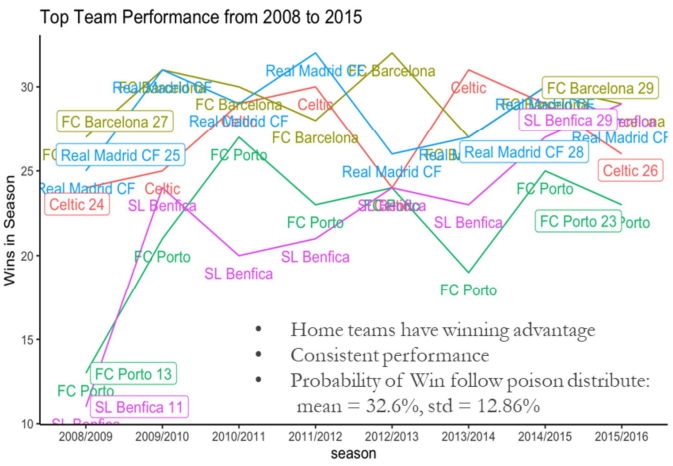
Feature Engineering for following tables is performed -

Match table: Each match can be home or away. A new table is created to have separate rows for home as well as away matches for same team. We need to find the characteristics of both teams involved to be able to learn about the Match. Therefore we create the features indicated above like goal scored, wins, losses, winning percentage, points and rank using the collective data of a season. Using these features given two teams, example rank 1 team and rank 10 team with relevant characteristics, our model is going to predict who will win the match. The heatmap, show the correlation among the features. There is strong correlation between loss and goals conceded which means, the more goals opponent scores against you the more you are likely to lose. Similarly on the opposite end the goals cored and the wins has strong correlation.

Player Table: We Feature Engineered few new columns in the Players table. A player can play in multiple positions throughout his career, so we added a column called "Total Positions" to judge versatility of a player. We also had a "Contract" column which we subdivided into "Contract Start Date" and "Contract End Date" and also calculated the number of years remaining in a players Contact. There were only player positions in the dataset, so to distinguish players created a "Category" column having four categories – Defender , Goalkeeper, Midfielder and Forward based on player position. Apart from that added few columns to predict value of player such as goals scored per appearance, minutes played per appearance and also the previous year value of the player.

Classify Team's Win/Lose





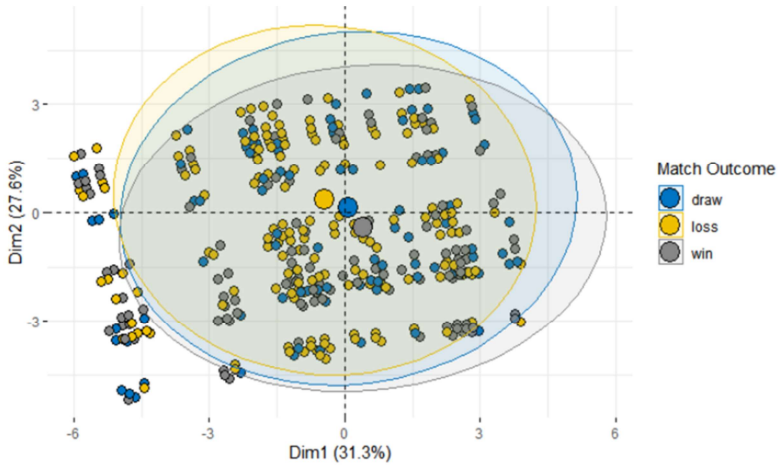
After we got the aggregated data and before we start to build the model, we would like to see what the common factors are to contribute to winning a match. We find in most sports, especially for soccer, being a home team can increase the probability of winning significantly. From the chart on the left, we can see on average, all the team have a 10% higher chance to win the match being a home team vs away team.

On the right chart, we would like to see the performance consistency for teams. Considering that we built a team performance table, including aggregated data for the team base on their yearly performance, we would like to confirm that team's performance throughout the seasons are consistent and our variable can contribute to the Machine Learning models. We find that majority of the team have high consistency throughout different seasons like shown in the chart, high-performance teams have 20 to 30 winning matches in each season from the year 2009 to 2015. And all the team's winning probability follows a poisson distribution of mean at 32.6% with standard deviation at 12.86%. We also created aggregated summation of each player's score (from 0 to 100, 100 being best), but visually there is not much of difference since the result is in range of 650 to 700 and home team player score summation has no visual difference than away team player score summation. But we will test this out in our model in the process of variable selection.

Classification Approaches

Excellent Failure: Close overlap of Draws with Wins and loses makes it hardest to predict.

PCA on the Match dataset:



Classification problem is solved using two different strategies.
a) Strategy 1: Use previous years data, plus current year to engineer predictors
b) Strategy 2: Use current years data only to engineer predictors

The features used in both the strategies are different since they use different approaches, however the features are selected using the feature/subset selection methods and checked for the relevance before using them in the model.

Here we can see from the PCA that the draw matches, severely overlaps with the wins and the losses. Running the model on the original dataset, we see the above graph after running PCA, which shows that draws had the highest misclassification rate. Perhaps the features of the teams are not able to differentiate them as draws. So we are excluding the matches that ended as draws in our model.

In Strategy 2, we are trying 2 methods of Model. One type is to use models without feature selection and in the second type we are using models that use feature selection method. The idea is to compare which models perform better.

Model Selection: Approach 1

Model & Data Set Selection	Accuracy
Logistic for 50% train, 50% test	69.44%
Logistic for 60% train, 40% test	69.54%
LDA for 60% train, 40% test	69.47%
QDA for 60% train, 40% test	69.09%
KNN for 60% train, 40% test	61.72%
QDA for 75% train, 25% test	69.02%
Random Forest for 60% train, 40% test	66.52%
SVM for 60% train, 40% test	69.51%

Bootstrap	Original	Bias	Std. error
t1	-0.81	2e-03	0.17
t2	0.86	-1e-04	0.02
t3	-0.0001	-5-e06	0.0002
t4	-0.0002	1-e06	0.0002
t5	0.007	2-e05	0.0014
t6	0.099	-1e-05	0.04
t7	-0.012	6e-05	0.02
t8	-0.08	-1e-04	0.004

In the first approach, we decided to transform the home team and away team out, replace with the 1st team and 2nd team and an extra home indicator indicating whether the 1st team being a home team or not. This way we can abstract home information out and fit it into our model to make a prediction.

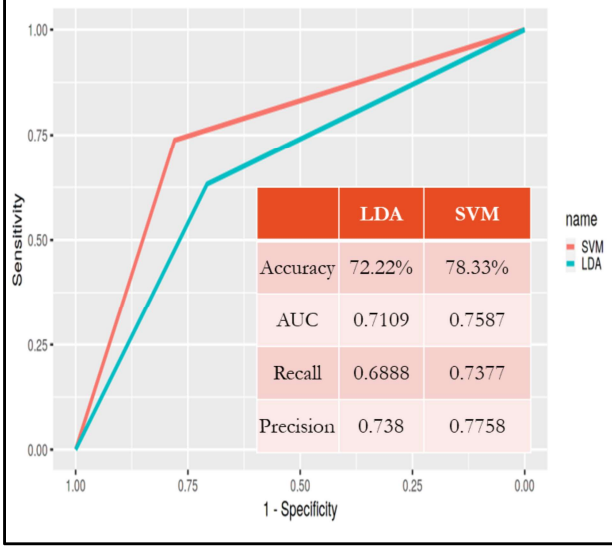
With the backward variable selection, we bring 12 selected variables down to 7 variables. To ensure our variable is reliable, we fitted our prediction variable into bootstrap for 1000 iterations and the result shows that all 7 variables have low bias and standard error.

We then tried to find a perfect balance to separate our training and testing data, the results show that even we use 7 years of data as training data, to predict last season result, the accuracy would still stay the same, due to our model rely on the team's previous year performance, increasing the size of training data does not have expected effect on improving the accuracy for prediction result, even decrease the performance like shown for QDA model. We thus choose to proceed with a 60/40 ratio for training and testing data. We tried all different kinds of Supervised Machine Learning modeling, including logistic regression, LDA, QDA, KNN, Decision Tree and SVM. Logistic regressions show the best performance but only with nuance difference than SVM with only 0.03% higher accuracy rate.

Model Selection: Approach 2

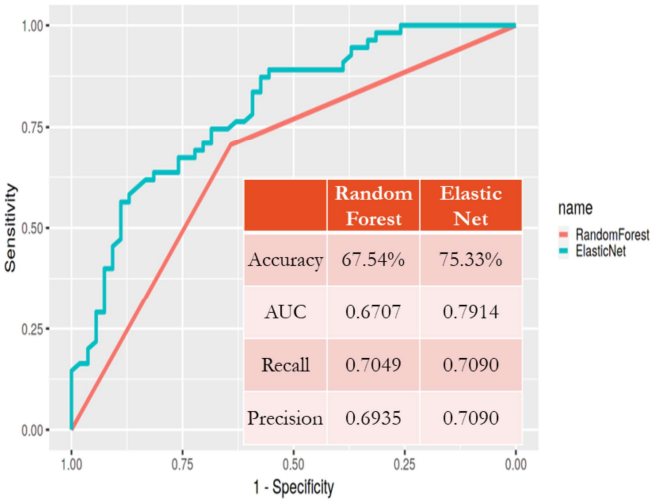
- Split : 80 % training , 20% testing

Using Non-Feature selecting Models



	LDA	SVM
Accuracy	72.22%	78.33%
AUC	0.7109	0.7587
Recall	0.6888	0.7377
Precision	0.738	0.7758

Using Feature selecting Models

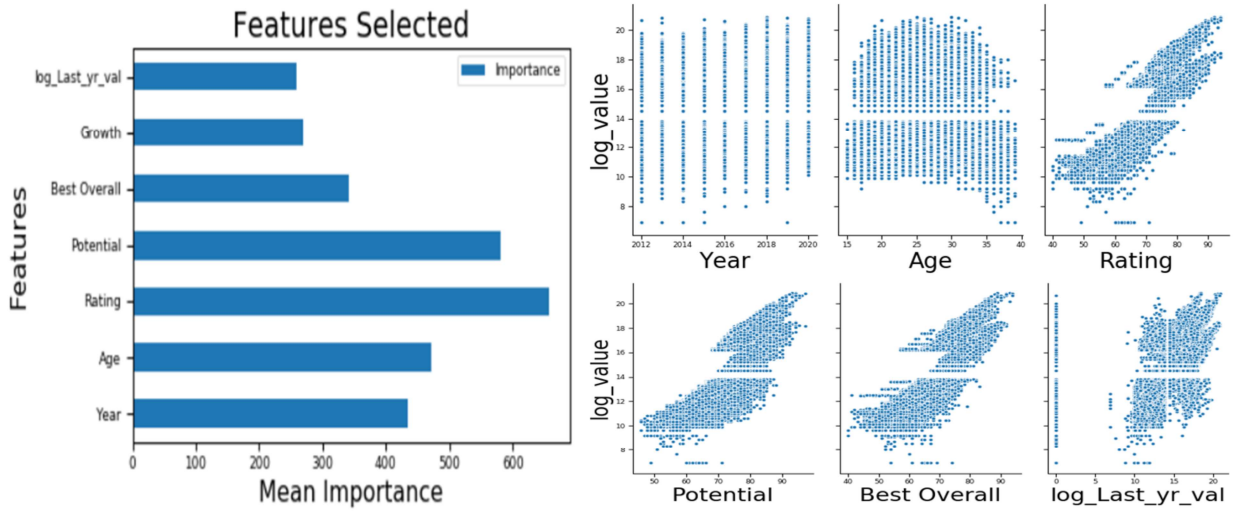


	Random Forest	Elastic Net
Accuracy	67.54%	75.33%
AUC	0.6707	0.7914
Recall	0.7049	0.7090
Precision	0.6935	0.7090

LDA and SVM methods are non-feature selecting, these models are fitted with features found after removing the correlated features which were found using Pearson Correlations (Feature selecting models like forward step function also gave similar features). The Random Forest and Elastic Net selected more features, and is able to generalize better for unseen data.
SVM and LDA tried to find a hyperplane that best separates the classes, using different approaches. We observe that both SVM and LDA approximate a similar Plane, however SVM uses the maximum variance between the features to obtain a hyperplane which produces a better fit than LDA which assumes the inputs to be Gaussian Distributed. Elastic Net, produces the best fit, as it addresses "over-regularization" by balancing the penalties between LASSO and Ridge. Elastic Net is able to classify using a better mix of features compared to SVM and LDA, as the feature selection is directed towards improving the classification. Random forest, comes up with random mix of features which provides the best gain, however since Elastic net uses more variables keeping some of the correlated variables, hence it is able to generalize better for the unknown data. Performance of Elastic Net relies on the training set. The model is able to generalize better since it keeps more correlated features, however if the training split is same as random forest then the performance is similar.

Value Prediction

- Initial Regression Model : 33% Accuracy.

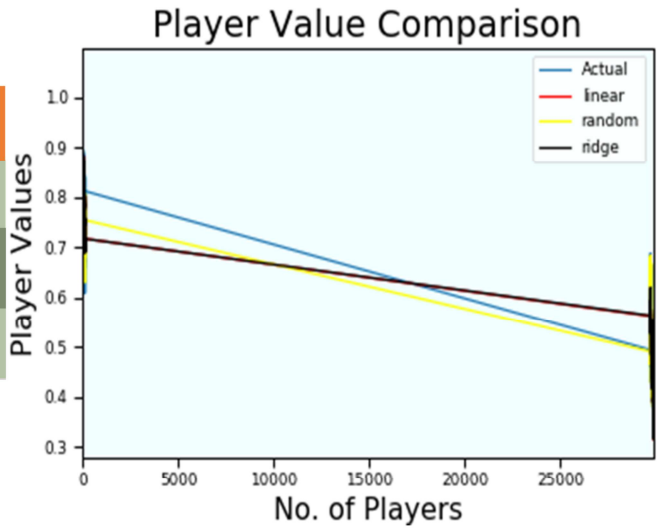


We are trying to predict the value of a player for the next season. We used different Feature Selection techniques such as Recursive Feature Elimination (RFE), LassoCV, Random Forest Feature Importance and Light Gradient Boosting Feature Importance. All the Feature selection techniques except RFE gave almost the same features. Recursive Feature Elimination gave around 62 optimal features. All the other techniques gave the features which are shown in the left figure on the slide. After running each Feature Selection technique, we tried to fit a Multiple Linear Regression model to predict the value based on the features selected. But after each technique we got an accuracy of only around 33% which clearly showed that there is not a clean linear relationship between the predictors and the target variable. We plotted a pairplot of the target variable with the common predictors and could clearly see that most of the predictors such as Rating, Potential, Best Overall, Year have an exponential relationship with the target variable Value. We took the log value of the target variable and the corresponding pairplot is the second plot in the slide. This plot tells us that there is now a linear relationship between the target variable and the predictor variables. We could also see from the plot that taking the last year value of the player also has an impact on current value. Apart from that, the general trend in age shows that value of a player decreases with the increase in age.

Model Comparison

- Split : 80 % training , 20% testing

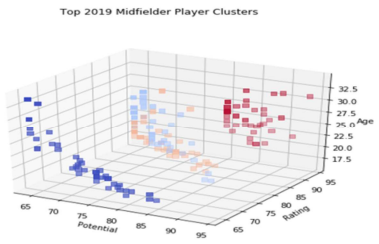
		RMSE (Training)	R2 (Training)	RMSE (Test)	R2 (Test)
1	Multiple Linear Regression	0.063	0.821	0.070	0.785
2	Ridge Regression : best lambda = 0.3	0.063	0.823	0.070	0.786
3	Random Forest	0.032	0.944	0.044	0.909

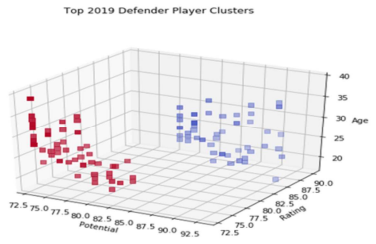


After obtaining the features, we split the dataset into 80% training and 20% testing data. We did Multiple Linear Regression by using the above features and got an accuracy of around 82% on the training and 78.5% on testing data. The F-statistic has a high value of 72740 which shows that the model was able to explain the variability in the data. Looking at the coefficients, we saw Ratings has a huge impact on the value which is expected as high rated players generally earn more than other players. Also, we could see that the coefficient for Age was negative which showed that with increase in age the value of a player decreases. We then performed Ridge regression and used cross-validation to get best value for lambda parameter which came out to be 0.3. The RMSE and R2 values for Ridge regression are almost the same as Linear Regression which tells us that there was no overfitting of data in Linear case. Also, Linear Regression was able to explain the Bias-Variance trade-off as Ridge Regression did not penalize even the most variable feature. To try for better accuracy we fitted Random Forest which gave us 94% accuracy in training and 91% accuracy in testing. We then plotted a line plot showing the actual and predicted values of players. The x axis shows the number of players in the dataset sorted by the ranking in descending order. We see that random forest predicts slightly lower value for players with higher ratings but the value predicted for lower rated players is very near to the true value. Ridge and Linear regression do not predict values well for very high and low rated players.

Player Clusters

Player Category	Clusters
Midfielder	4
Forward	4
Defender	2
Goalkeeper	3





- Example of Player's replacements:

ID	Name	Year	Category	Value	Age	Rating	Potential
153079	S. Agüero	2019	Forward	645000000	30	89	89

ID	Name	Year	Category	Value	Age	Rating	Potential
202126	H. Kane	2019	Forward	965000000	24	90	92
188545	R. Lewandowski	2019	Forward	77000000	29	90	90
188567	P. Aubameyang	2019	Forward	59000000	29	88	88
194765	A. Griezmann	2019	Forward	71000000	27	89	89
231747	K. Mbappé	2019	Forward	82000000	19	88	95

ID	Name	Year	Category	Value	Age	Rating	Potential
167948	H. Lloris	2019	GoalKeeper	36000000	31	88	88

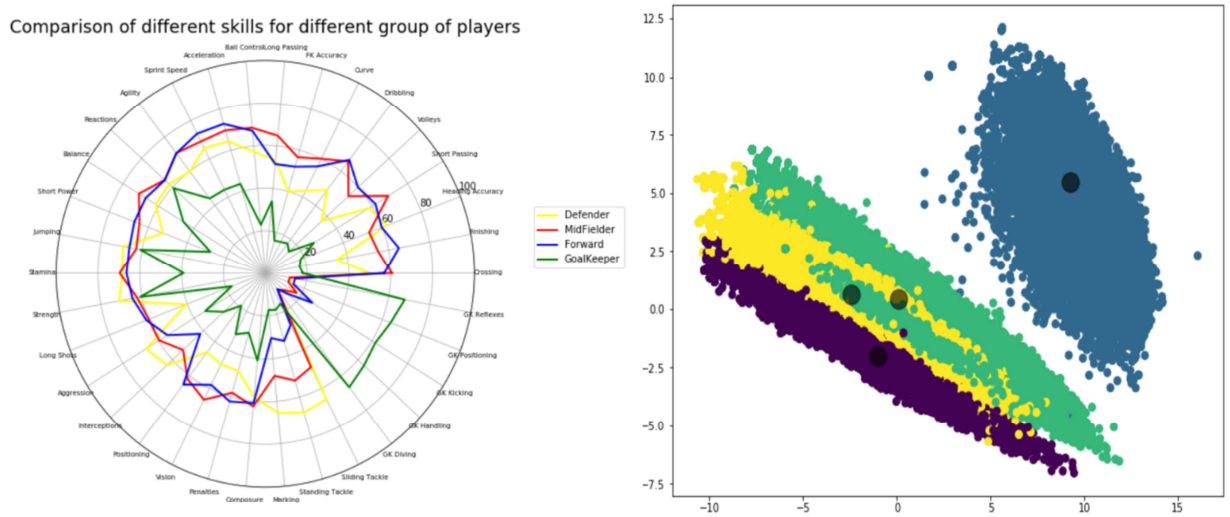
ID	Name	Year	Category	Value	Age	Rating	Potential
193080	De Gea	2019	GoalKeeper	625000000	27	90	92
192448	M. ter Stegen	2019	GoalKeeper	58000000	26	89	92
192119	T. Courtois	2019	GoalKeeper	475000000	26	88	89
212831	Alisson	2019	GoalKeeper	46000000	25	87	91

We used K-means Clustering technique for each categories and using the Elbow Method we determined the number of clusters in each categories. Top rated players of each clusters in a particular category are represented in 3D plots with Potential, Ratings and Age as the axis. For Defender category clusters are seen easily separable, while for Midfielder category two clusters are overlapping as the players are clustered using 53 attributes, representing higher dimension clusters in lower dimension results in overlapping.

Example for replacement of players:

- Forward player: S. Aguero (Value: 645M, Age: 30, Rating: 89, Potential: 89)
- We got 5 similar players with different attributes and values. The player K. Mbappe has value: 82M, Age: 19, Rating: 88, and Potential: 95, this might be best replacement as the value is much less and with the age of 19 he can play for more years than any other player as the average retiring age in soccer is around 34.
- Goalkeeper player: H Lloris (Value: 36M, Age: 31, Rating: 88, Potential: 88M)
- We got 4 similar players with different attributes and values. Although all the players we got have more value than the given players, but all of them are in their mid 20's. So even if the Club Manger have to pay more for the replacement it will be worth as they can play for more years than the given player.

Clustering Players



Clustering of similar players will help Team Manager to compare players attributes and their values also helps to find the replacement of a player if the player gets injured or leave the club. It also helps Club Manager to build an all round team in a limited budget. In our player dataset players are distributed in 4 categories i.e. Midfielder, Forward, Defender, and Goalkeeper.
We created a Radial plot for comparison of attributes of these 4 categories and it showed that Midfielder, Forward, Defender have almost similar attributes and Goalkeeper have different attributes. At first we tried to cluster players across all the categories using K-means Clustering and Gaussian Mixture Model techniques.

After doing PCA and applying Gaussian Mixture Model we found that attributes of Midfielder, Forward, Defender are almost overlapping while, Goalkeeper have different set of attributes and skills. Therefore, we decided to separately cluster players according to 4 categories.

Summary

- More features does not necessarily improve accuracy of predicting winner of a Match
- With the answers to our objective questions, a Manager can make changes to the team before a match, replace the players leaving the club based on suggestions at an appropriate value.
- **Upsets**: Matches where a low ranked team beats a high ranked Team
- **Elites and Outliers**: Players who earn considerably more than the their peers
- We aim to improve our project further by improving the features through research and training the Model using more data from other relevant resources.

We had many numeric and categorical features differentiating the home and away team, however none of the categorical features were significant and did not improve the Model, and only a subset of numeric predictors improved the Model. Similarly only a few predictors out of 80 odd predictors were relevant in getting a good estimation for players value.

The psychological impact of losing one match could influence a team to continue to lose few more and so on, in spite of having a good team. These factors are currently not in our model as they are hard to measure and quantify. Due to these parameters soccer is a unpredictable sport and getting high accuracy is very difficult.

Injuries, frequency of the matches in a week, player's off field behaviour among other thing impact players performance which are currently not captured in the Model.
In addition to this, few players are worth much to the team than their performance indicates. Hence this is very difficult to model. The predicted player value even though is analytical, sometimes may not be equal to the true value. One example of this could be a player who has a huge fanbase but is not in good form may still earn a lot more than others due to the revenue that he might bring to the club.