Predicting the Market Value of FIFA Football Players

I. Problem Statement: Why is this useful?

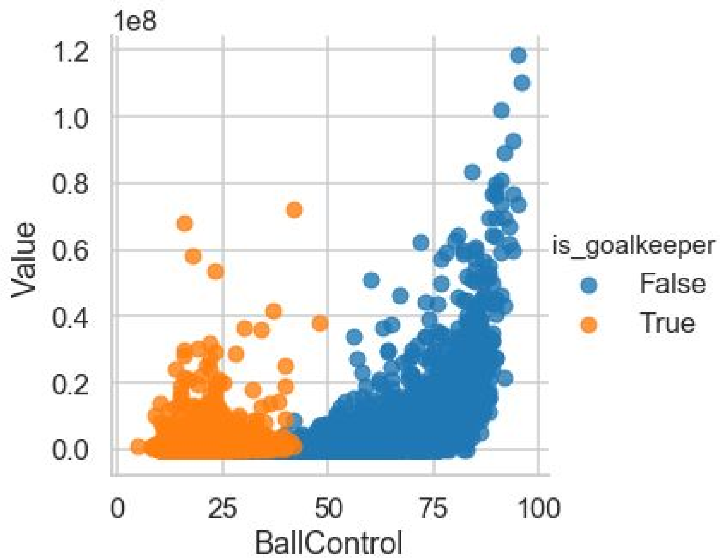
The term “Moneyball” was coined back in 2002 when the Oakland Athletics built a baseball team of undervalued players based on statistics. You may have even seen the film called “Moneyball”. This strategy is especially important for clubs that have lower budgets. Football clubs pay millions to purchase football players. A Model that can predict a player’s value would help club managers make informed decisions on how much they should pay for a player with certain attributes. Additionally, such a model could help managers “bargain” players that would perform almost as well as top players. Now, whether or not these statistics do not apply to their real player counterparts, this model could still be useful for managing fantasy football teams.

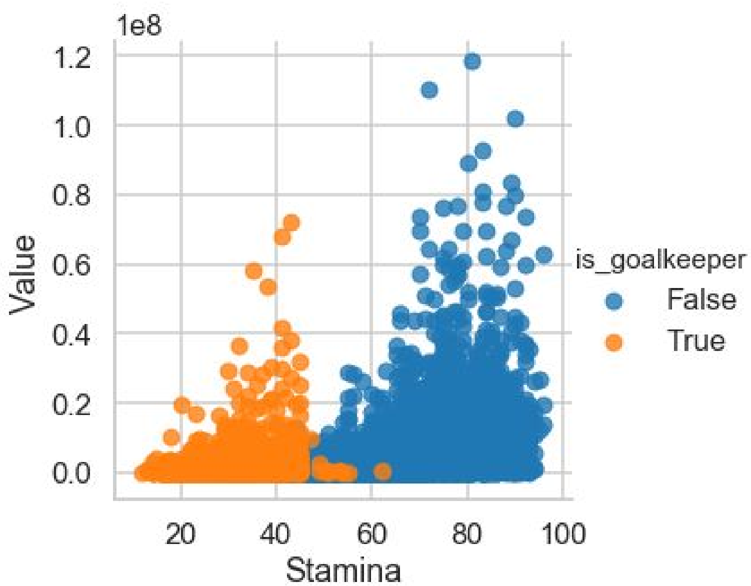
II. The Dataset: How was obtained and prepared?

I obtained the FIFA19 dataset from Kaggle, which made the data cleaning and wrangling process much simpler. The entire process could be split into three parts: format adjustment, missing values, and removal of redundant attributes. Format adjustment portion pertained to removing spaces, converting values, and splitting columns. For example, I needed to change the very important Value and Wage columns from ‘100k’ format to ‘100,000’ format and convert heights to centimeters in order to have one uniform unit. I also converted the non-numerical weight column to a numerical one by removing the units(lbs). As far missing values go, I filled those with their mean, mode, or median depending on the attribute. I removed redundant attributes such as ID, Photo, Flat, Club Logo, Loaned From, Real Face, due to their repetitive nature. These are only some examples of the data wrangling steps that I took to prepare my data for exploratory data analysis(EDA). As I dove into EDA, I chose to remove more columns based on visualizations and to avoid problems with multicollinearity.

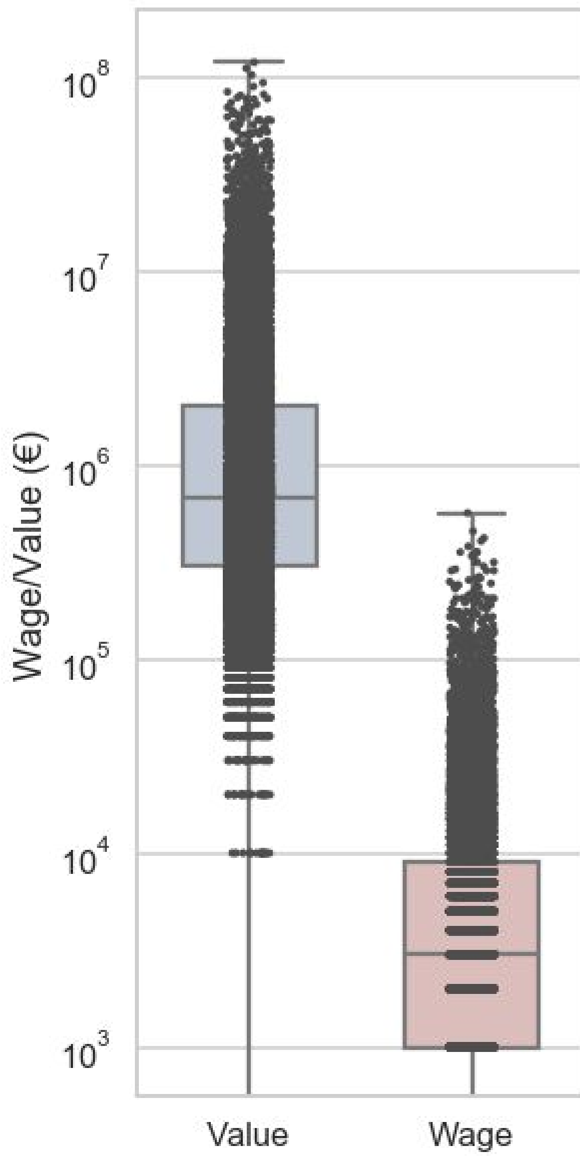
III. Exploratory Data Analysis: Interesting findings based on visuals and statistics

Since I would like my model to focus on field players rather than goalkeepers, I removed all the goalkeepers from the dataset. This will remove the amount of noise in my later analysis. Also, certain attributes--ball control, stamina-- are different for goalkeepers. Thus, removing goalkeepers from my data analysis will provide much more accurate results when predicting player value. Couple attributes for goalkeepers vs. field players are shown below. Demonstrates how the two groups are differently distributed.

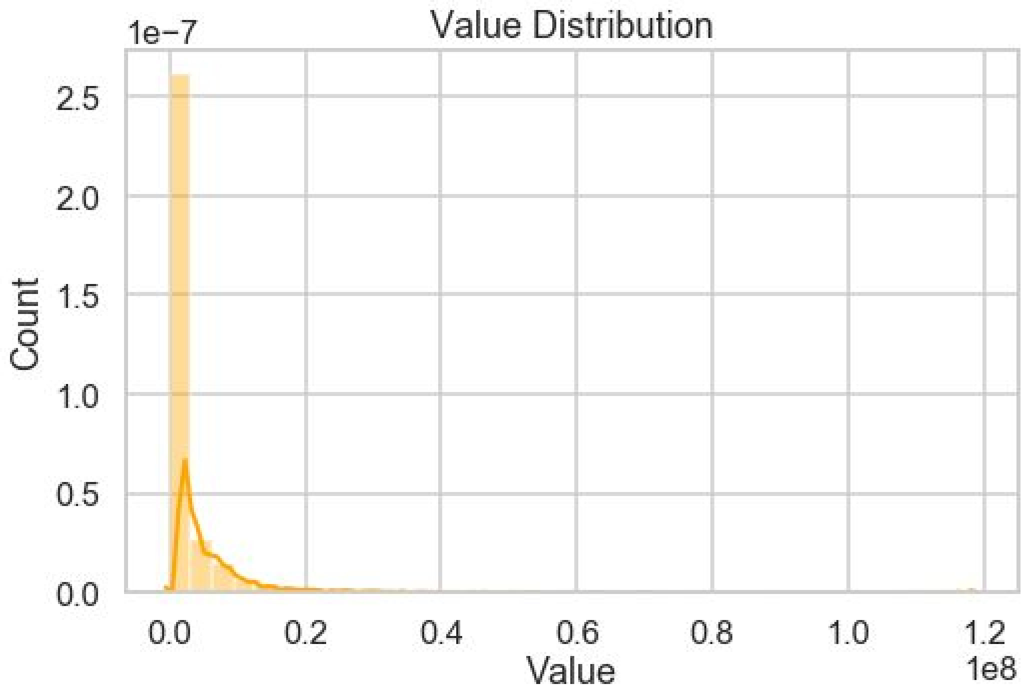




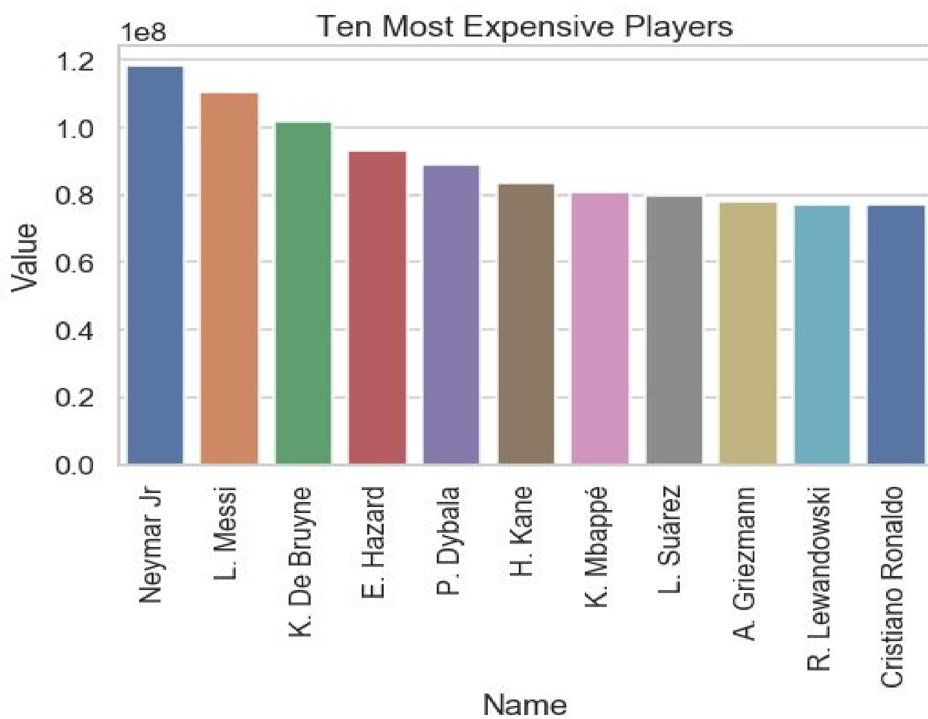
First, I explore my target variable : Value alongside Wage.

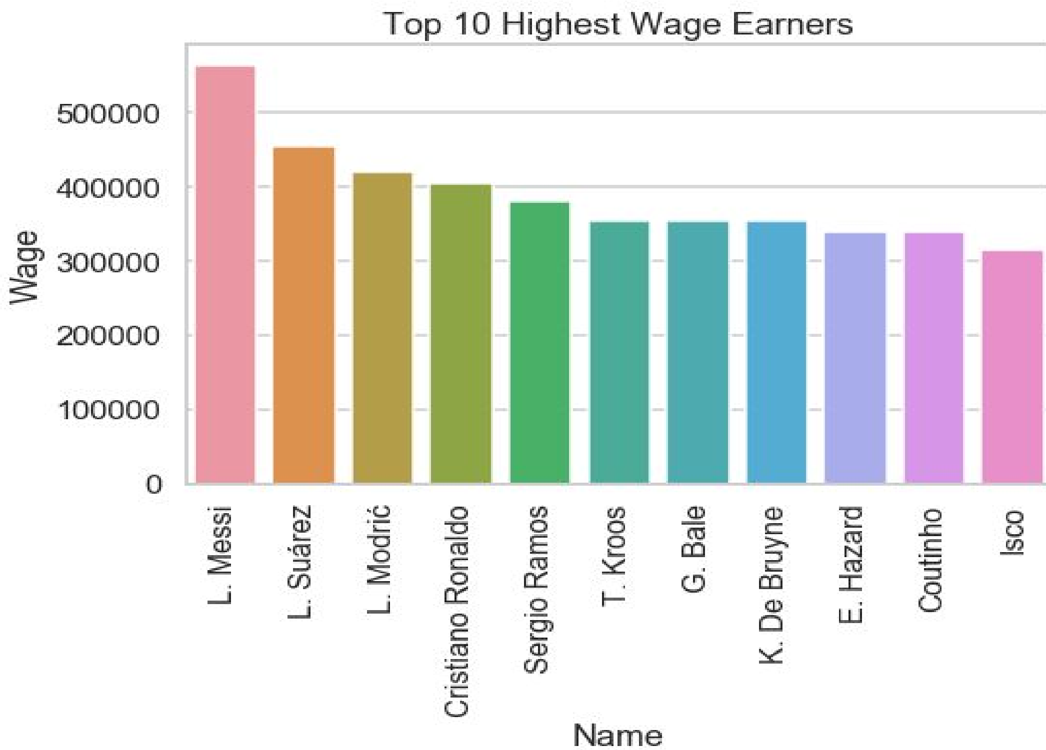


It’s clear that Value is highly skewed such that, we have many average valued players, then we have the superstars who are valued significantly higher than the rest. Is this because they are that much better? Or perhaps, the slightest advantage in certain skills, results in much higher value. Below is the distribution for Value to get a clearer picture of how right skewed it is.

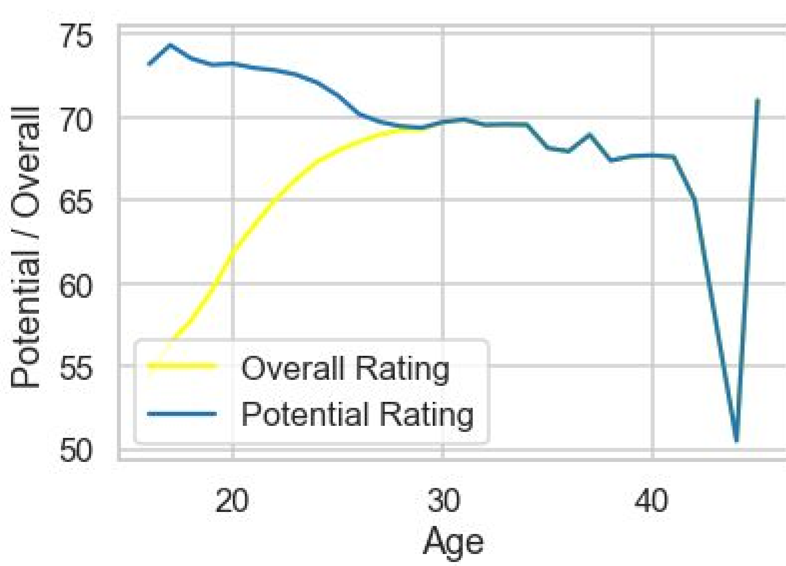


We can see here that, most players fall into one bin, as the value increases, there are less and less players in the respective bin. This is expected in the sport of football. Following are the most valued players.

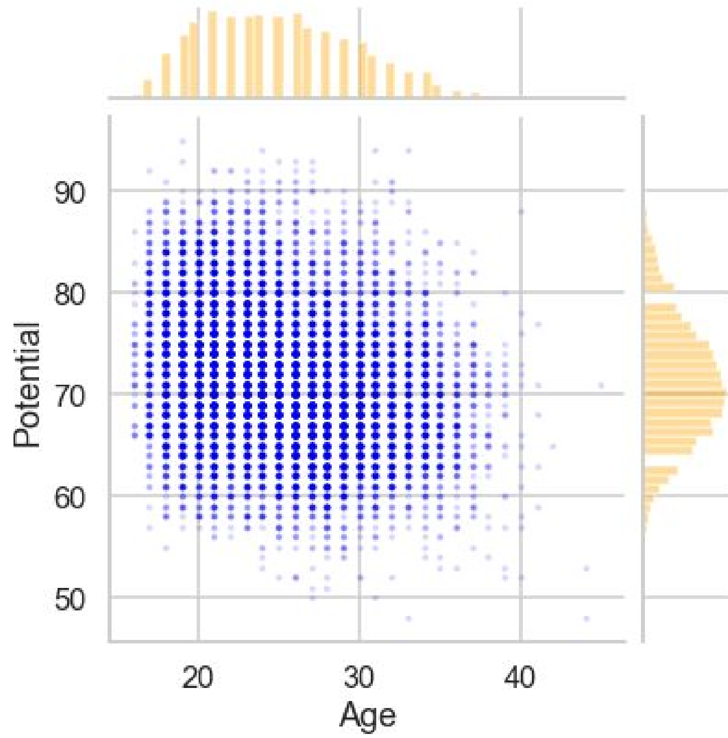




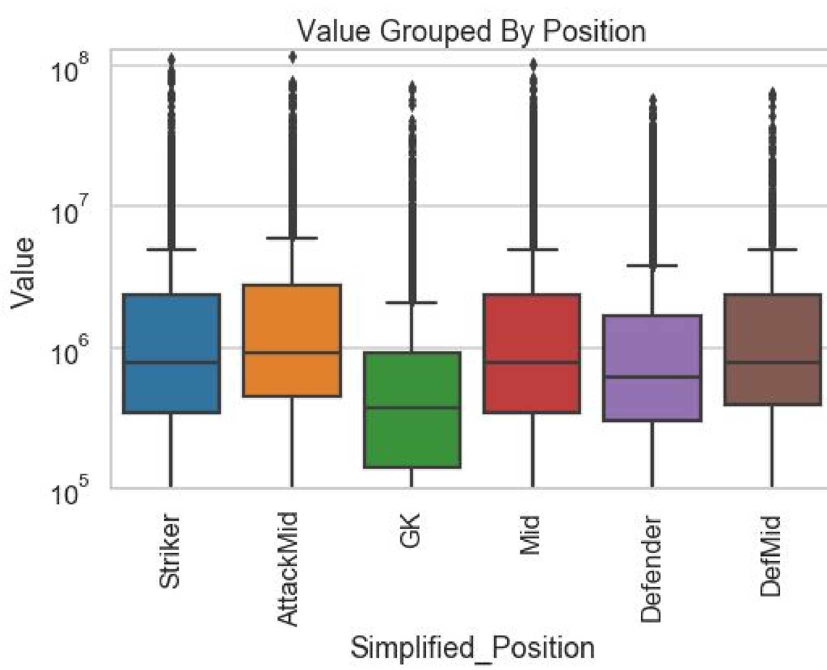
We can make an observation that highest valued players aren't necessarily the highest paid players. Although, these variables do have a very high correlation. Surprising to not see Neymar Jr. here as he is the most valued player.



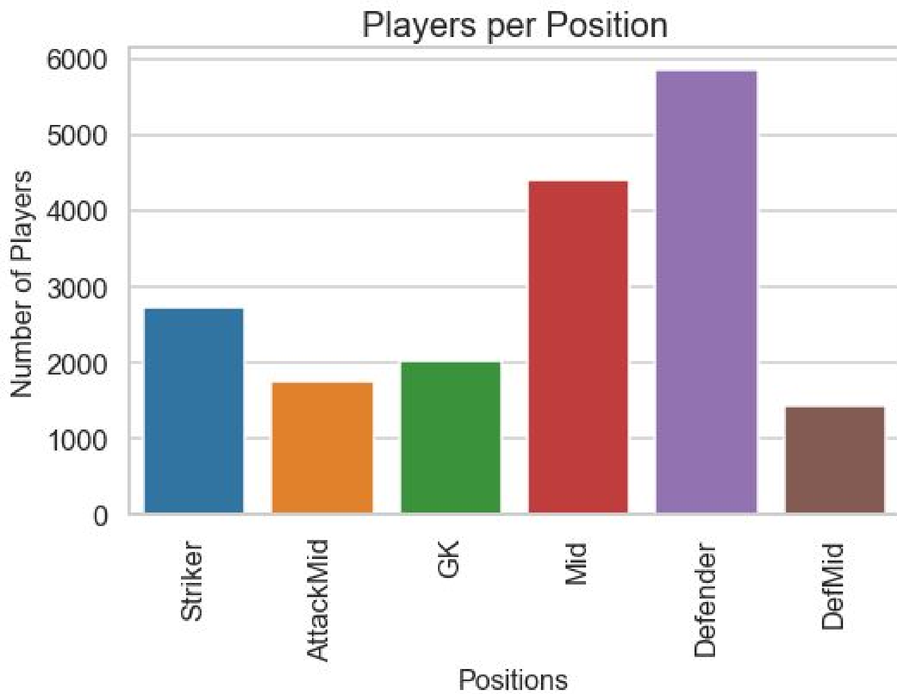
Interesting observation between age and Potential rating (which is a proxy of the overall rating), is that players reach their highest rating at the age of ~29 years.



Let's see whether we can gain any insights about Value based on position. For this, I aggregated the different positions into six main positions: GoalKeeper, Defender, Defending Midfielder, Midfielder, Attacking Midfielder, and Striker. This will make it much easier to see if there are any significant differences.



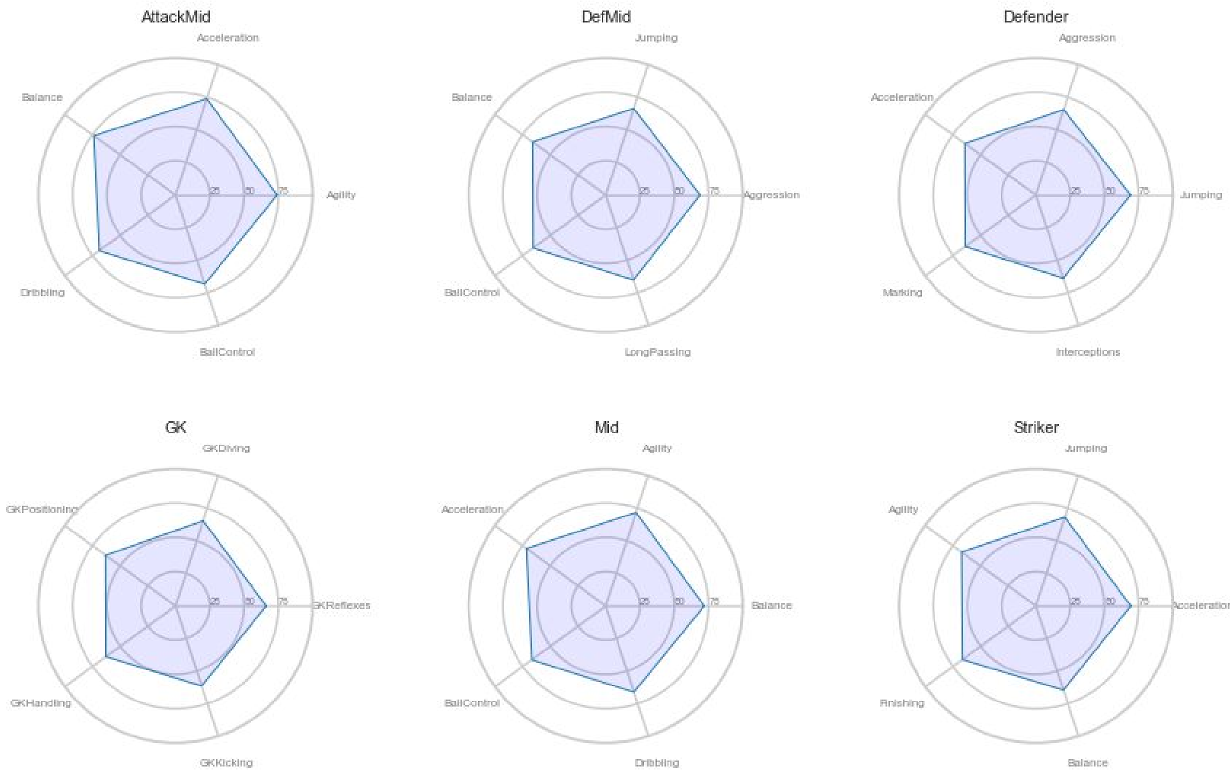
We can see that attacking types of positions are slightly paid more on the extremes. This slight amount could be significant given the amount of each player costs. Goalkeepers are paid less on average, but there are of course many outliers in this category as well who are valued just as high as any other position. While we're comparing positions, let’s also get an idea of which position has the most players.



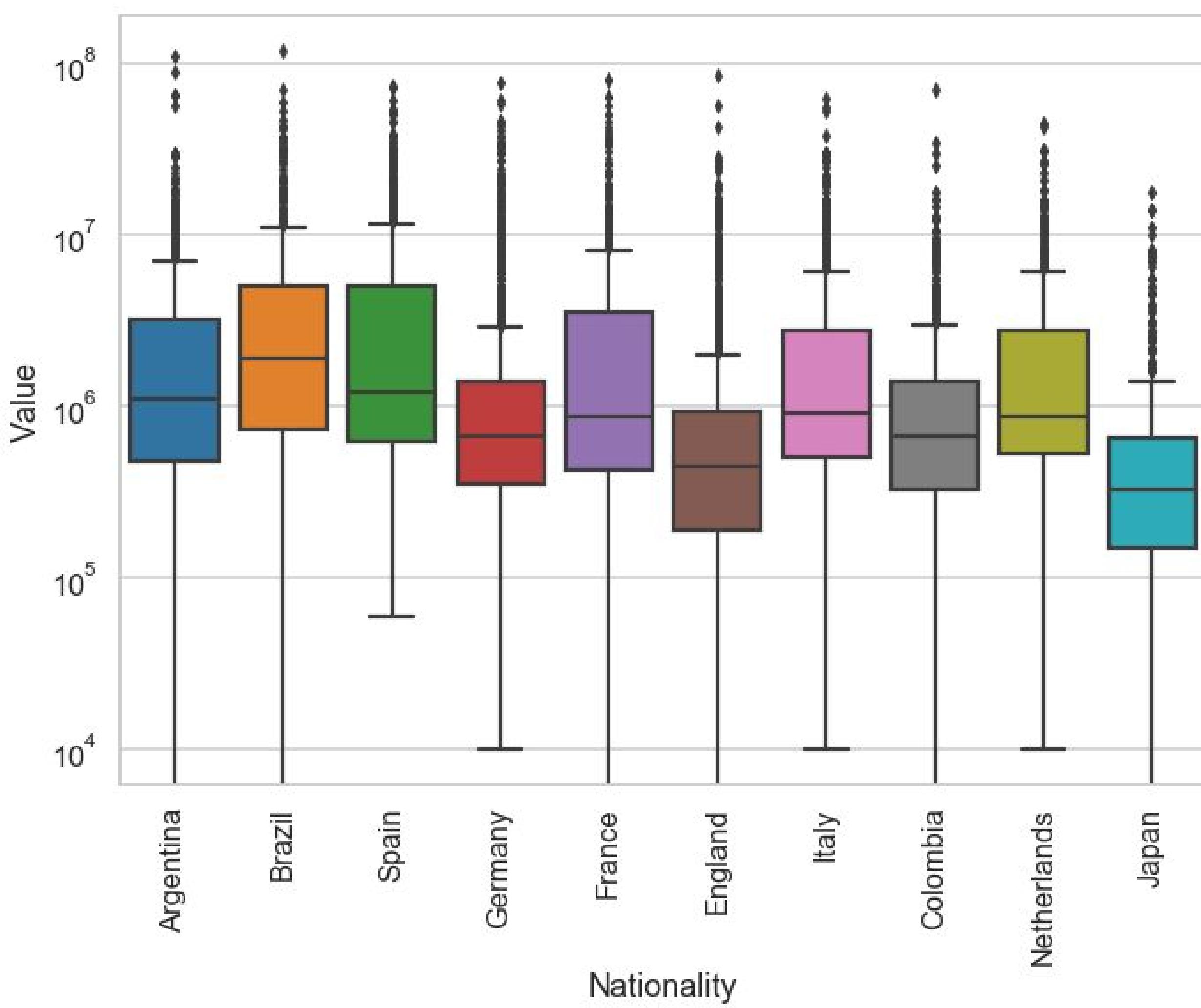
Perhaps, the abundance of defensive positions makes it so that they are not paid as much on average, even though the differences may be insignificant. Let’s take a closer look at the most dominant attributes for these positions.

We can see that different positions have different dominant attributes(See the next page for the figure). Therefore, we can say that the attributes in some way also define a player’s position. This is useful to know because I would like to predict player’s based on their attributes, rather than what position they play.

There are also some clear similarities between attacking positions such as strikers and attacking midfielders, as well as defenders and defending midfielders. So not only are they similar in terms of player Value, but also player attribute ratings.

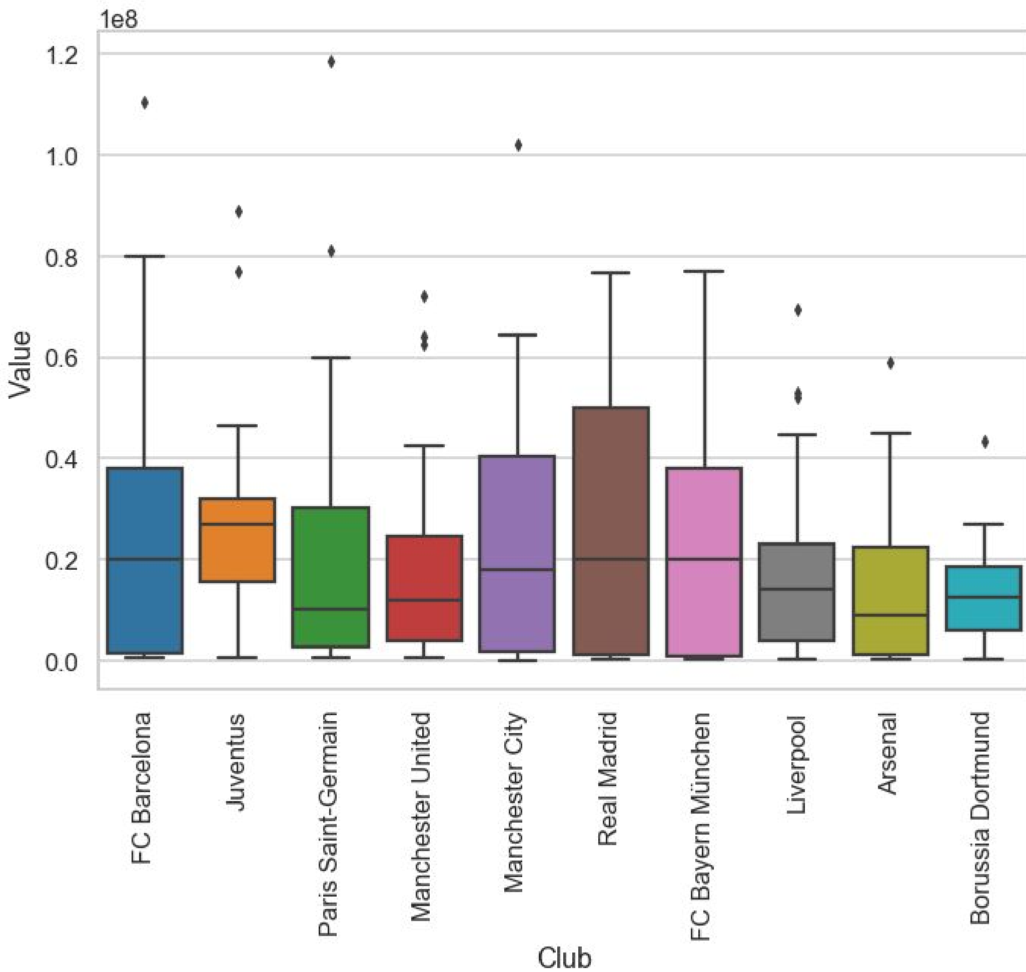


AttackMid
Acceleration
Balance
Agility
Dribbling
BallControl
DefMid
Jumping
Balance
Aggression
BallControl
LongPassing
Defender
Aggression
Acceleration
Jumping
Marking
Interceptions
GK
GKDiving
GKPositioning
GKReflexes
GKHandling
GKKicking
Mid
Agility
Acceleration
Balance
BallControl
Dribbling
Striker
Jumping
Agility
Acceleration
Finishing
Balance



Value
10^8
10^7
10^6
10^5
10^4
Argentina
Brazil
Spain
Germany
France
England
Italy
Colombia
Netherlands
Japan
Nationality

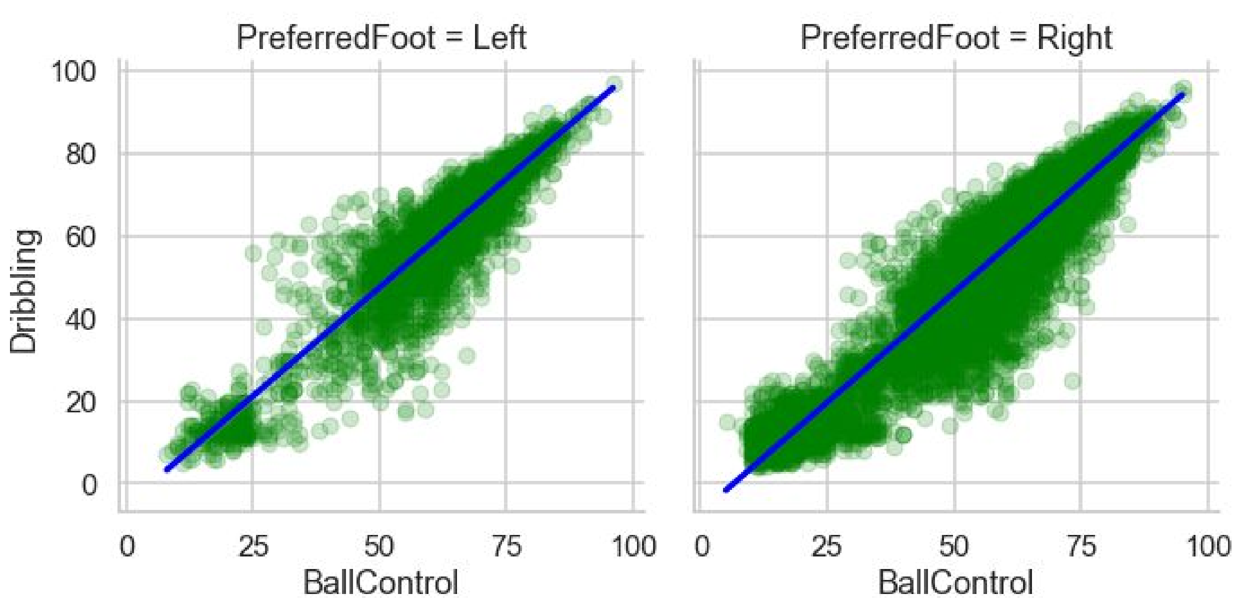
We can see that on average, players are valued the most from South American (Brazil, and Argentina) as well as Spain. Clearly, there are outliers for these top 10 Nations, but we can perhaps say that International Reputation may an important feature for our model. Let's take a look at the clubs as well.



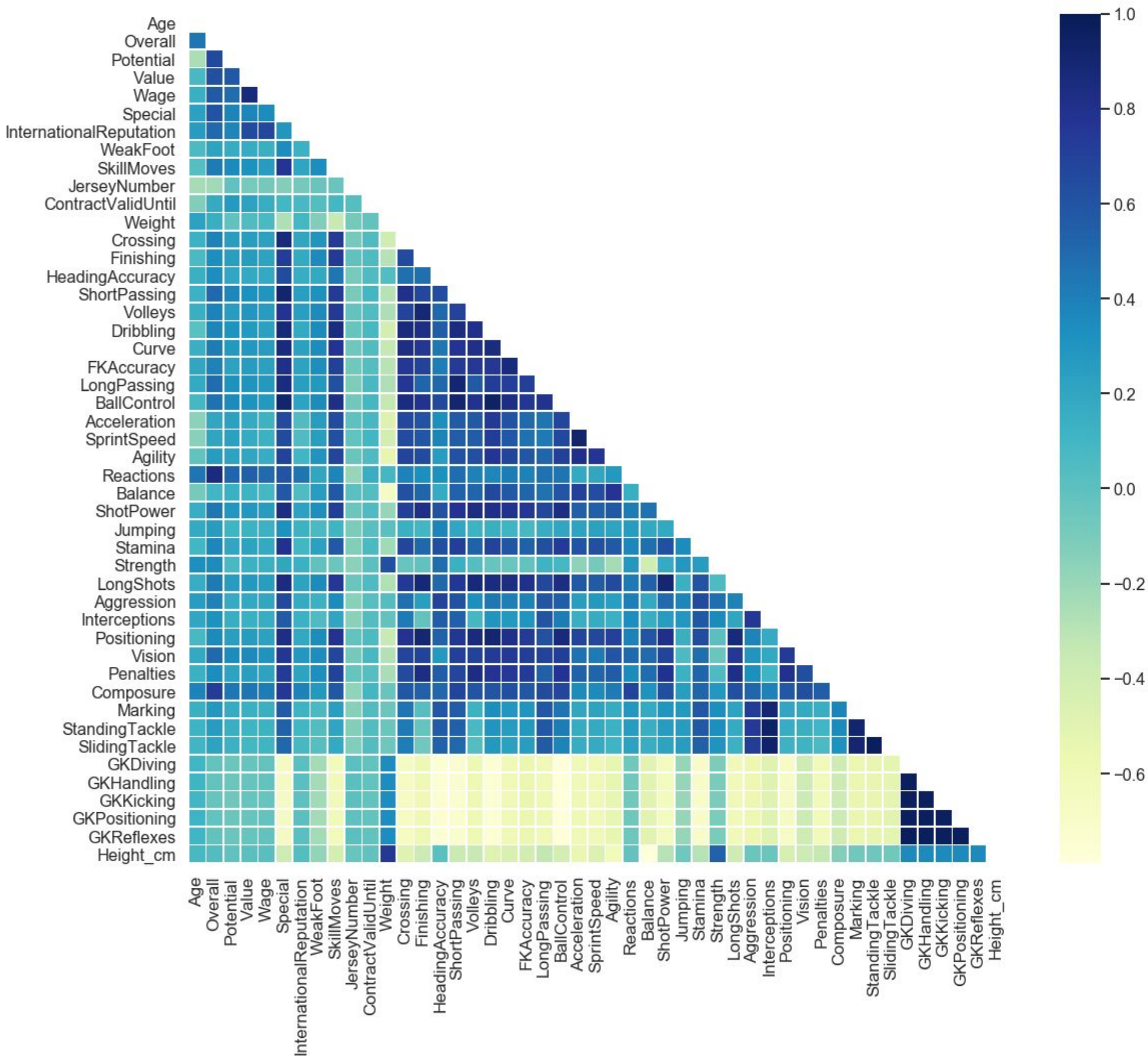
The clubs that consist of highest valued players both Spanish (Barcelona and Real Madrid). An interesting club here is Juventus as it has a very high median value (highest) of players but it's not as spread out as the other high valued clubs. Juventus seems to have this philosophy of hand picking players of the highest quality even though they may not be superstars like Messi of Barcelona, or Neymar from Paris Saint Germain. I've inserted a chart of the top 15 valued players and their nationality as well as the club they play for. It consists of players from either Europe or South America.

	Name	Value	Wage	Nationality	Club
2	Neymar Jr	118500000.0	290000.0	Brazil	Paris Saint-Germain
0	L. Messi	110500000.0	565000.0	Argentina	FC Barcelona
4	K. De Bruyne	102000000.0	355000.0	Belgium	Manchester City
5	E. Hazard	93000000.0	340000.0	Belgium	Chelsea
15	P. Dybala	89000000.0	205000.0	Argentina	Juventus
16	H. Kane	83500000.0	205000.0	England	Tottenham Hotspur
25	K. Mbappé	81000000.0	100000.0	France	Paris Saint-Germain
7	L. Suárez	80000000.0	455000.0	Uruguay	FC Barcelona
17	A. Griezmann	78000000.0	145000.0	France	Atlético Madrid
10	R. Lewandowski	77000000.0	205000.0	Poland	FC Bayern München
1	Cristiano Ronaldo	77000000.0	405000.0	Portugal	Juventus
11	T. Kroos	76500000.0	355000.0	Germany	Real Madrid
30	Isco	73500000.0	315000.0	Spain	Real Madrid
31	C. Eriksen	73500000.0	205000.0	Denmark	Tottenham Hotspur
3	De Gea	72000000.0	260000.0	Spain	Manchester United

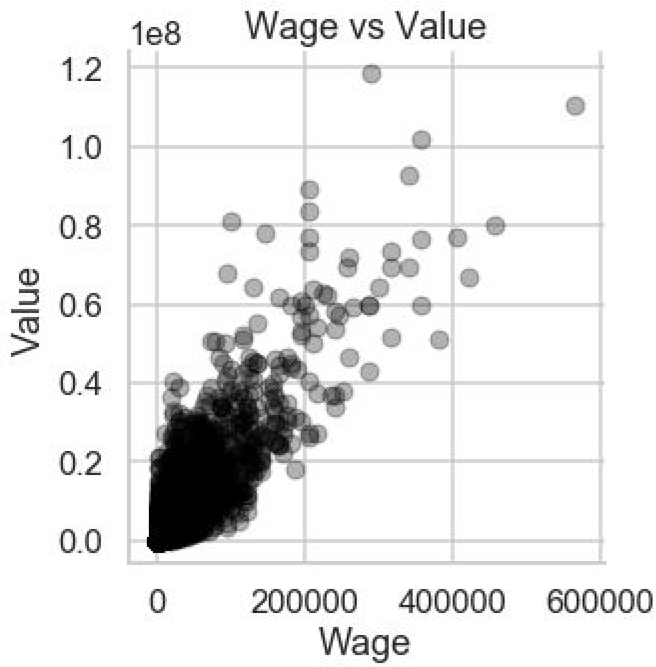
What about foot preference? Does that have any impact on important football attributes such as Ball Control or dribbling?

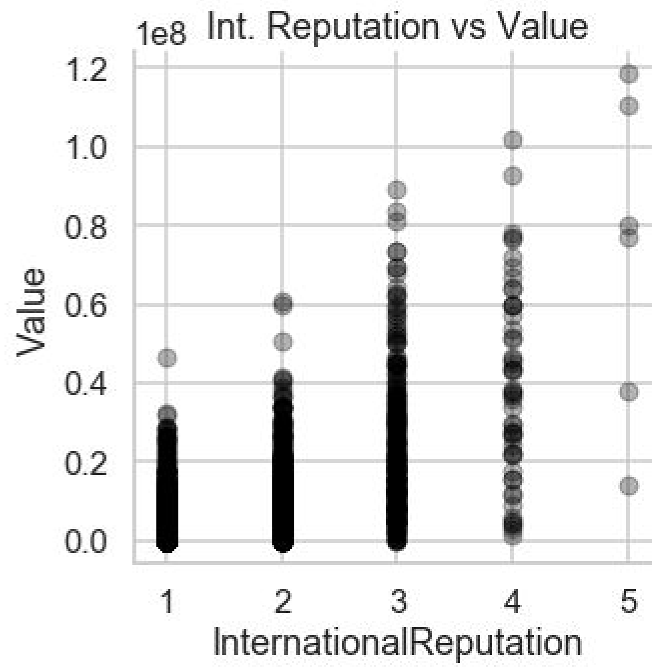


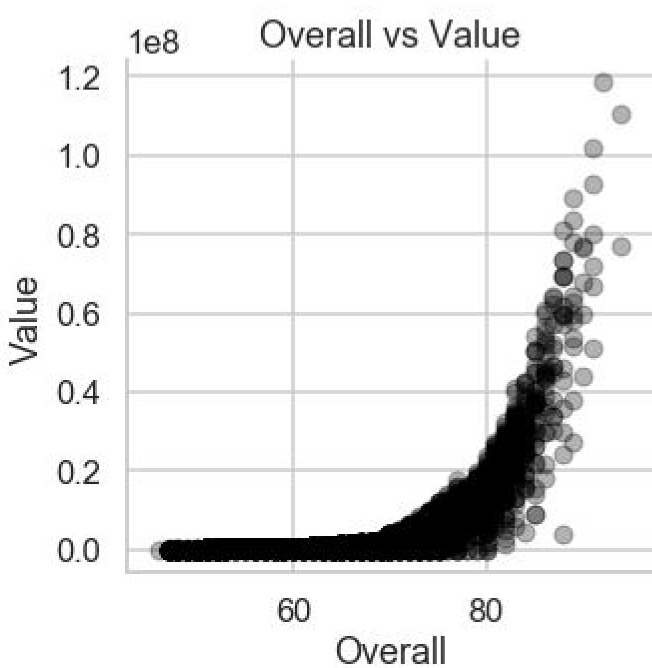
We can see that foot preference has no effect on ball control or dribbling, other than the fact that there are more right foot players than left footed ones. We will likely exclude foot preference from our list of attributes that will be used later on for machine learning. Let's take a look at all the variables from a bigger scope, using a heatmap. This will give us insight about any linear relationships between our predictors as well as our target Value. This will help with choosing attributes in a way such that we can minimize or ideally completely avoid multicollinearity.

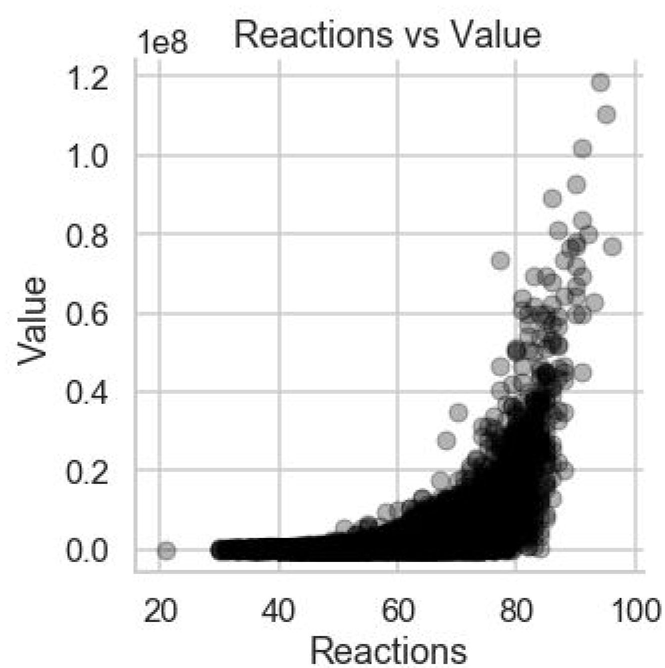


Let’s consider any Pearson correlation coefficient value (r) above 0.5(absolute value), to be of significance, and will require further exploration when selecting a model for predicting player’s value. Let’s take a look at the top 4 predictors based on correlation: Wage, International Reputation, Reactions, and Overall rating.









We can see for all the variables here that they have a positively strong correlation with player value: The correlation values are as follows : Wage(0.86), Overall(0.63), Int. Reputation(0.65), and Reactions(0.54).

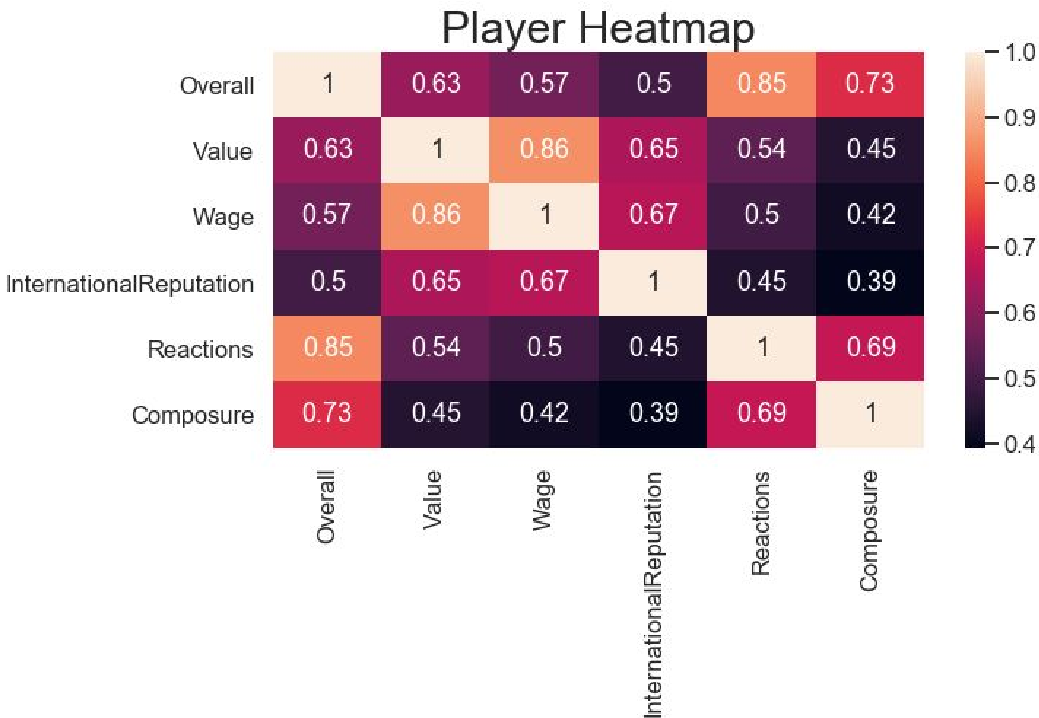
Before, doing any Linear regression, I shall first look at the ANOVA table with different size models in order to see what variables do best when you consider them simultaneously.

We have the following ANOVA results for the model *Value ~ Wage + InternationalReputation + Reactions + Overall + Composure,* based on the highest correlation values.

```
                             df        sum_sq       mean_sq             
F  \
Wage                        1.0  4.199774e+17  4.199774e+17  58447.534
094
InternationalReputation     1.0  6.468290e+15  6.468290e+15    900.180
882
Reactions                   1.0  7.604239e+15  7.604239e+15   1058.268
916
Overall                     1.0  4.920042e+15  4.920042e+15    684.713
770
Composure                   1.0  1.533055e+14  1.533055e+14     21.335
266
Residual                18201.0  1.307841e+17  7.185545e+12
NaN

                                PR(>F)
Wage                      0.000000e+00
InternationalReputation  4.396260e-193
Reactions                1.081045e-225
Overall                  3.438348e-148
Composure                 3.882125e-06
Residual                           NaN
```

We can see that all of the p-values for the 2-way ANOVA are close to 0. Thus, if our null Hypothesis is that at least one of the estimators is 0, we would reject that hypothesis. Meaning, all of these features add significant value to this initial Ordinary Least Squares model. Let’s also take a look at the heatmap zoomed in at the features above.



We can see that Composure has the lowest correlation with our target “Value”. Also, we can check for multicollinearity here between any of our features. I will only consider correlation coefficient values above 0.7 for multicollinearity. We can see that Overall has high correlation with Reactions and Composure. There is also moderately strong correlation between Wage and Reputation, as well as Reactions and Composure. We may consider dropping Composure or Reactions or both, if we see that the model performs better.

IV. Machine Learning

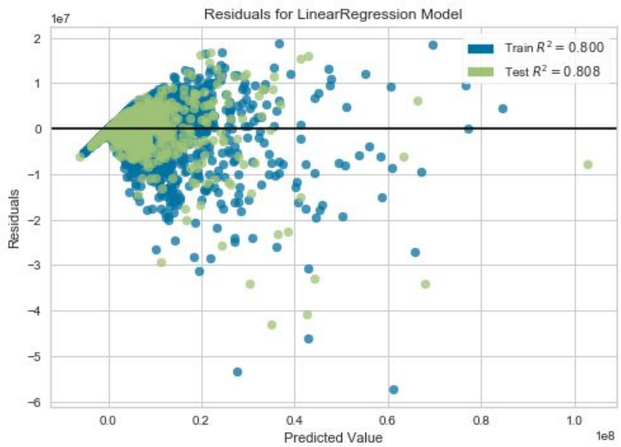
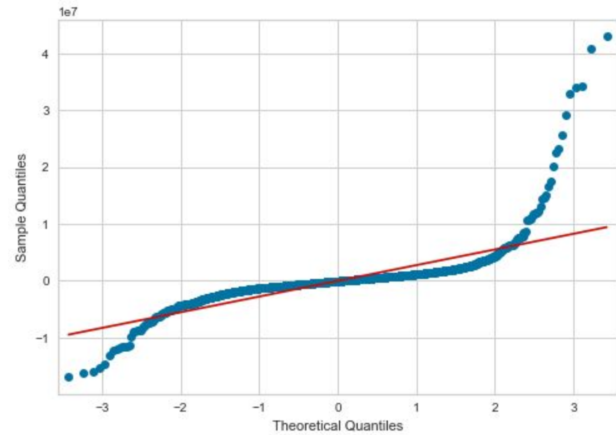
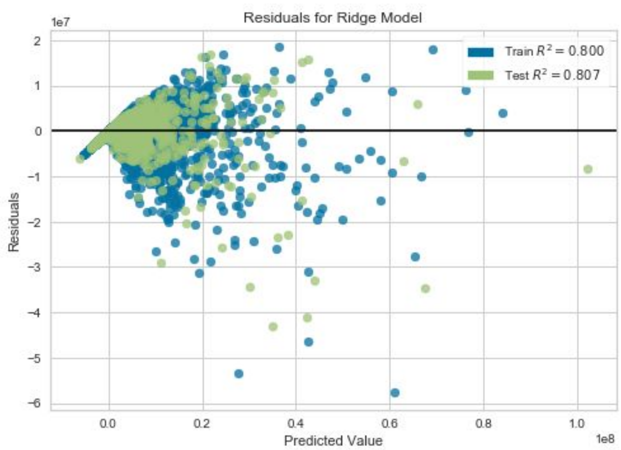
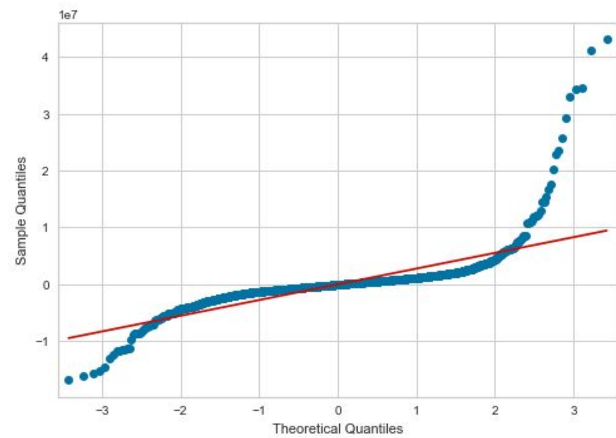
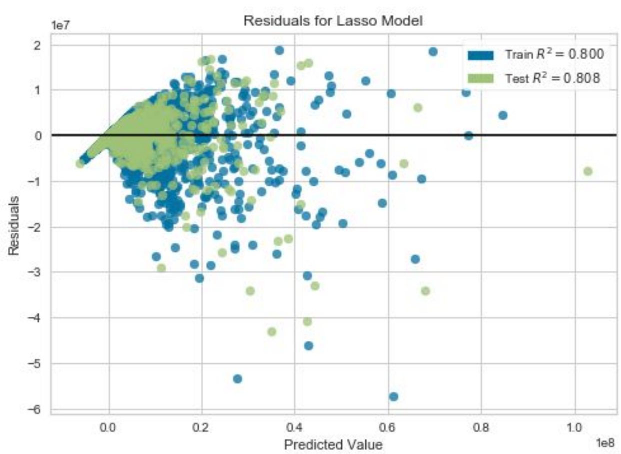
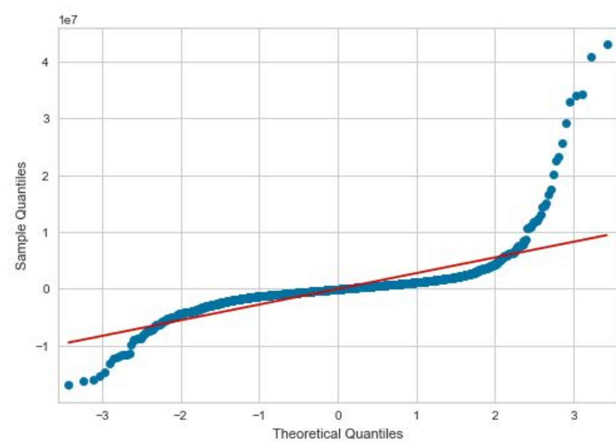
For this portion I will be evaluating each model based on Root Mean Square Error (RMSE - Root mean square deviation) as well as checking for overfitting by comparing Test and Train sets’ results. I will also examine the residuals to ensure they meet model assumptions if necessary. For future improvement I will tune hyperparameters for the model of my choice based on this preliminary examination.

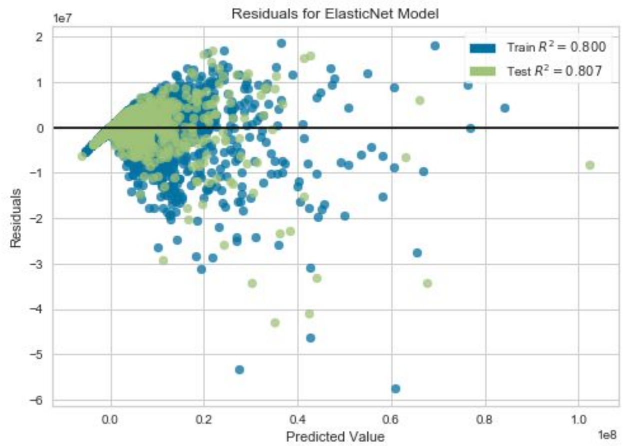
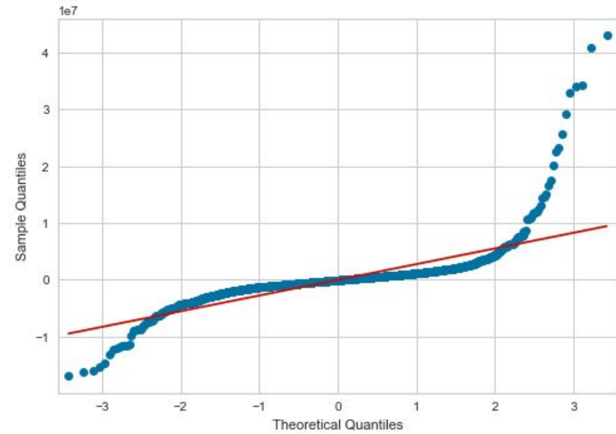
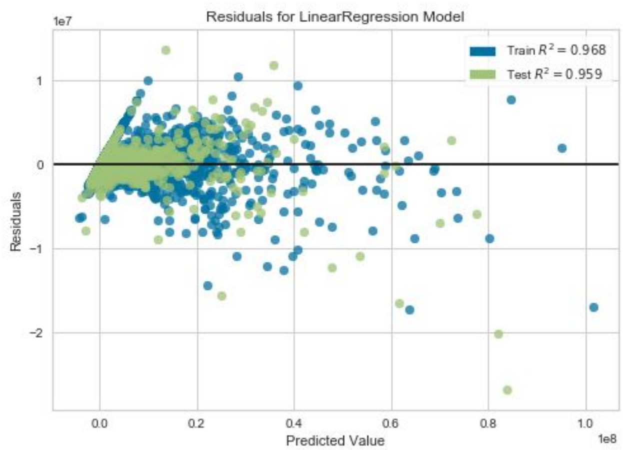
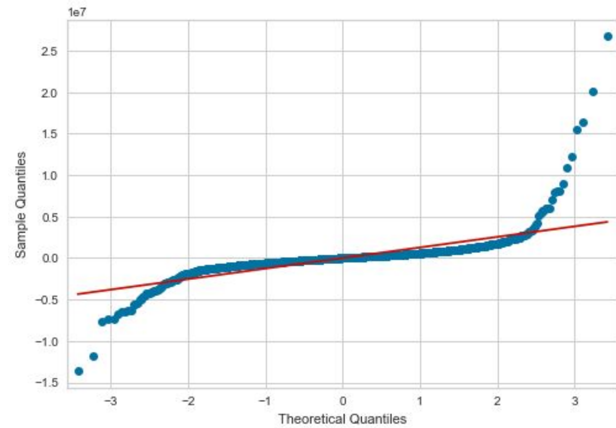
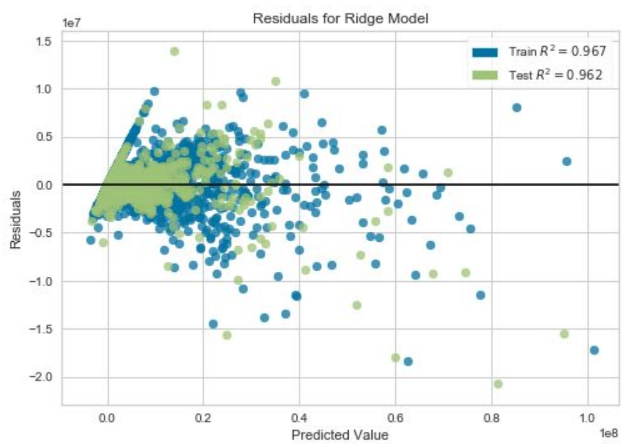
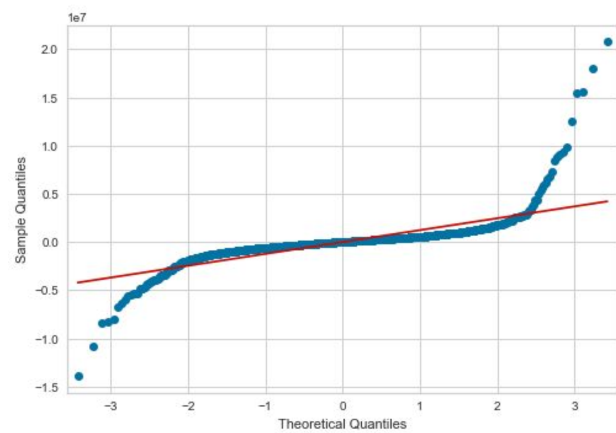
RMSE : Lower = better

Results for the base models :

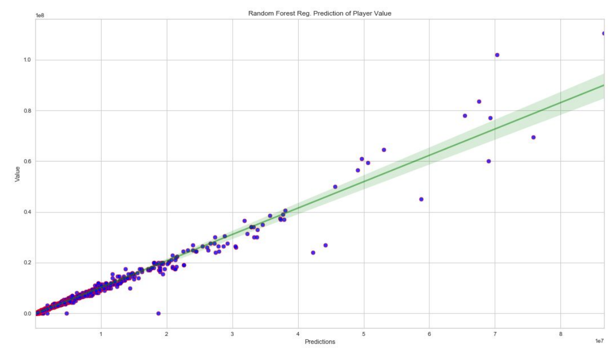
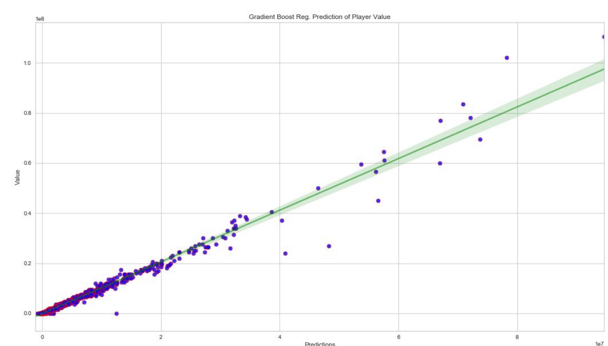
Model	R-sq	RMSE	Train Set Score	Test Set Score
OLS Linear Regression	0.808	2761137.789	0.800	0.808
Ridge Regression	0.807	2765402.455	0.800	0.807
Lasso Regression	0.808	2761562.436	0.800	0.808
ElasticNet Regression	0.807	2763822.351	0.800	0.807
Polynomial OLS Regression(deg. = 2)	0.959	1271424.785	0.968	0.959
Ridge Reg. (deg. =2)	0.962	1230446.037	0.967	0.962
Random Forest Regression	0.966	1154656.079	0.997	0.966
Gradient Boosting Regression	0.979	951692.694	0.997	0.970

We can see that as we go down the list of models, we see lower and lower RMSE scores. However, once our RMSE lowers significantly (starting with Polynomial regression), we can see possible issues of overfit. Hyperparameter tuning as well as feature tuning could help alleviate some of these concerns. Since my data’ residuals do not follow the assumption that residuals must be identically and independently distributed (Normally), I will choose to go with the ensemble models. This is a more efficient solution than trying to adjust the data such that the assumptions are met.

Model	Residual Plot	QQ Plot
OLS Linear Regression		
Ridge Regression		
Lasso Regression		

ElasticNet Regression		
Polynomial OLS Regression (deg. = 2)		
Ridge Reg. (deg. =2)		
Random Forest Regression	Does not apply	Does not apply
Gradient Boosting	Does not apply	Does not apply

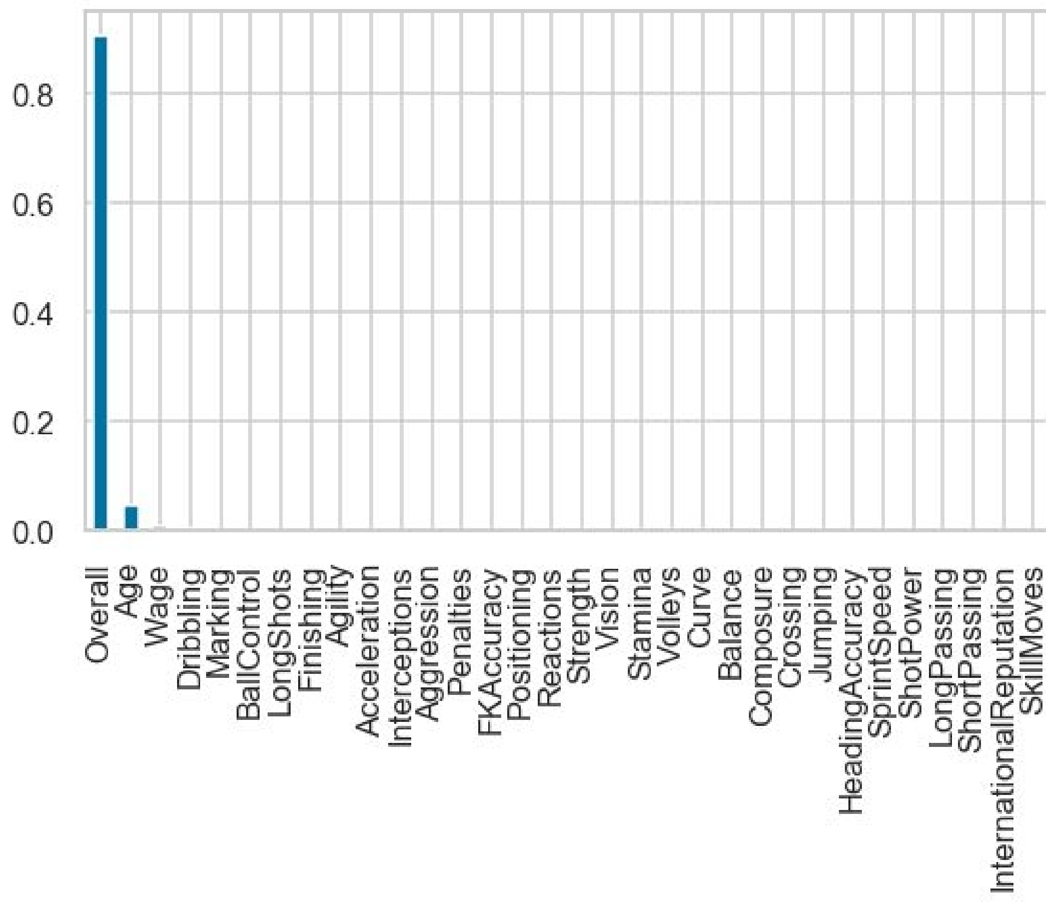
As far as residual plots go, we see an issue of residuals not being completely univariant and random. This could be due to overfitting as well as multicollinearity. The residuals' variance does go down as we go down the list of our models. QQ plots on the other hand seem to have an issue of outliers, which is why the beginning and the end of the plots significantly deviate from the theoretical values.

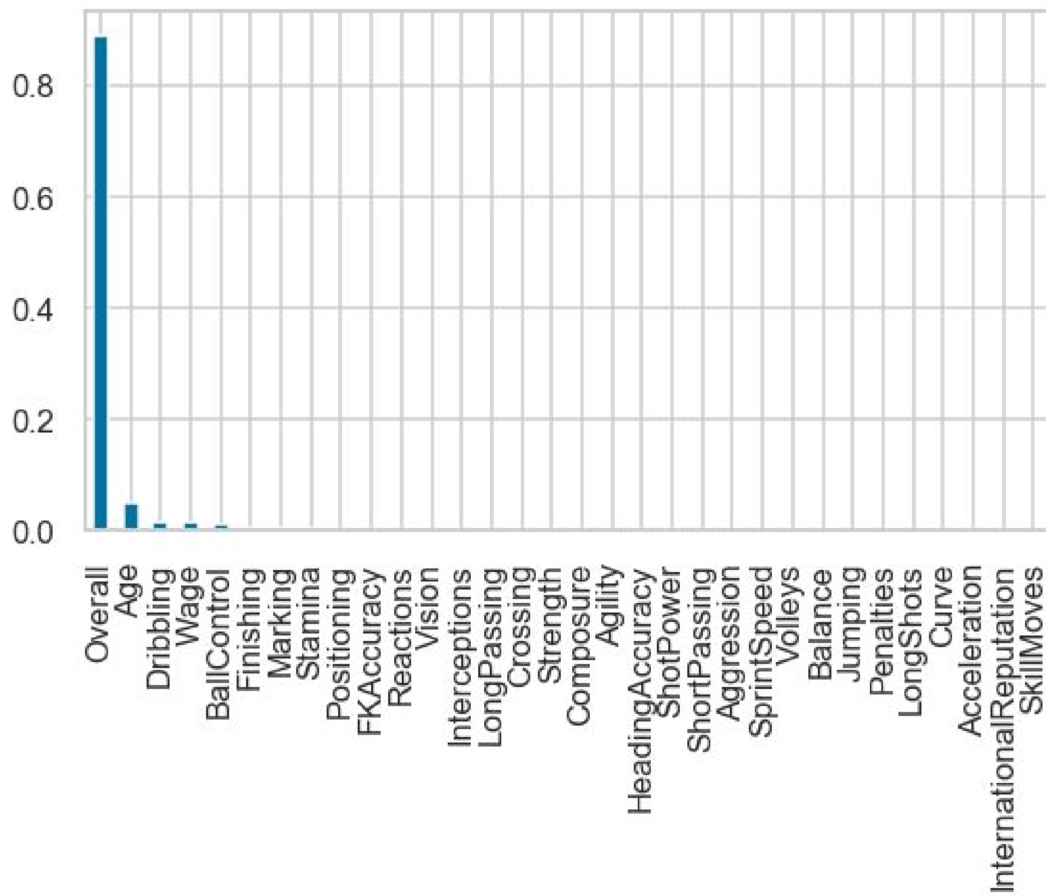
Model	Actual Values vs. Predicted Values
Random Forest Regression	
Gradient Boosting Regression	

Performance of these models is very close, however, the gradient boosting model has a slight advantage when we compare the RMSE.
After tuning the hyperparameters with cross validation I received the following results for my metric of evaluation: RMSE.

Model	RMSE	RMSE(base)
Random Forest Regressor(RFR)	1115881.201	1154656.079
Gradient Boosting Regressor(GBR)	922059.417	951692.694

There is a slight improvement in both models after tuning the hyperparameters.

Model	Feature Importance
RFR	

GBR	

We can see that the random forest model performs slightly worse, however, it only has 2 significant features. Gradient boosting regressor performs better and it puts a bit more significance on 3 more features.

Conclusion - Findings and Recommendations

The goal of this project was to create a model that was able to predict a player's value (i.e. is he worth the money) based on simple linear regression. This could be helpful for clubs with lower budgets as it would allow them to organize a team of undervalued players. However, since these linear models require certain assumptions to be met, it was more beneficial to go with ensemble models after examining the residuals for several linear models including ridge, lasso, polynomial regression. I attempted to remedy this by using only the most important features based on correlation; however, this did not change the variance of my residuals enough to use those models.

Given the ensemble models (random forest and gradient boosting), a player's overall score is the best predictor for player's value. This is no surprise, as overall value showed high correlation to player's value. However, factors like age and dribbling were not good predictors of player's value in my initial regression models involving OLS, Polynomial, etc. I was surprised to not see Wage as an important feature in the gradient boosting or random forest model, as wage and value are very much related to each other. This is good in a way because wage is dependent on the player's value where you get a chicken and an egg problem of which comes first.

One possible implementation of this model can be to create the best value team based on purchasing undervalued players. Since each player has all the features in my chosen models we can see whether they're undervalued by comparing their actual value with the model based value. We can do this for all the positions on the field. This was done in baseball by Oakland Athletics and proved to be very effective.

Future steps to improve my models could be trying to tackle the problem with residuals in the initial linear models. Given the simple nature of such a model, it would probably be more practical and intuitive. Certain remedies include transforming the data, adding interaction terms, or engineering new features to reduce multicollinearity.