DS 4400: Machine Learning and Data Mining 1

Spring 2021
Project Report

Project Title: Predicting Market Value of Soccer Players
TA: Omkar Reddy Gojala
Team Members: Jalaj Singh and Bennett Thorson
Code and Presentation (Google Drive):
https://drive.google.com/drive/folders/1FfAVNrsTnajHg9gxJg7O3IKSV11uSGyN?usp=sharing

Problem Description

Soccer teams across the world spend millions to hundreds of millions of dollars on transfer fees for players to join their teams. Unlike American sports leagues like the National Basketball Association (NBA) and the National Football League (NFL), soccer teams often acquire players from various leagues and countries, which creates a large pool of available players to choose from. Getting the right deal can make or break a team's season and can have an instrumental impact on the team’s financial operations. Our goal is to create a model that can predict a player’s market value using various player attributes (age, height, nationality, goals, etc.), with the hope of finding value deals for teams. This machine learning problem is a regression problem as the features will be used to predict a player's market value (continuous number) in dollars.

“The buying and selling of players is a critical component of soccer” [1]. Clubs invest heavily in their scouting infrastructure to attain the necessary players for their squad as well as value deals that could turn a large profit in the future. All teams have a budget which they work with each season and finding the right players for the right price is essential to both a successful season and the club as a business. Analysts are a new, but important, group involved in purchasing players through the transfer market [1]. Analysts explore data and statistics on players to help determine whether a player will provide value to the club and would be worth pursuing. Machine learning models, such as those described for this project, could prove beneficial for analysts in understanding if a player is overvalued by the selling club or undervalued and could prove to be a diamond in the rough for the buying club.

Similar projects that have been attempted include an article written on *Towards Data Science* where regression models were used to predict the market value of soccer players using data from the popular soccer video game FIFA [2]. Using Polynomial Regression, the author claimed an R^2value of 0.836 or 0.930 by taking the logarithm. The accuracy is respectable but the data raises concerns as this is data from a video game that is inferred from the developers and may not be reflective of the value and ratings of players in real life. Another similar project was performed by Sidharrth Mahadevan from Bournemouth University who scraped features from Transfermarkt, among other sources, to predict the market value of players using OLS Regression [3]. This project will follow a similar path in scraping data from Transfermarkt (more on this later), but will likely use different attributes and will utilize more regression-based models and feature selection techniques with the hopes of improving accuracy.

References

1. https://sites.duke.edu/transfermarket/introduction/the-transfer-market/
2. https://towardsdatascience.com/predicting-market-value-of-fifa-soccer-players-with-regression-5d79aed207d9
3. https://www.researchgate.net/publication/347439782_Predicting_Market_Value_of_Football_Players_using_Machine_Learning_Algorithms
4. https://www.investopedia.com/terms/w/wisdom-crowds.asp
5. https://www.sciencedirect.com/science/article/pii/S0377221717304332
6. https://projects.fivethirtyeight.com/global-club-soccer-rankings/

Dataset

https://www.transfermarkt.co.uk/ is global soccer's leading website for player transfers, player statistics, match information, and more. The website has information on approximately 800,000 professional soccer players. This information includes basic profile data, game statistics, medical history, contract duration, and importantly, the estimated market value for each player. Transfermarkt was built on the idea that users can estimate a player's market value similar to or better than football experts through a theory called "wisdom of crowds". This was proposed by James Surowiecki in his book "The Wisdom of Crowds" in which he theorized that the many are smarter than the few [4][5]. The market values on Transfermarkt are not calculated by the website. Instead, the values are estimated by aggregating the individual estimations of members of the community where these estimates are then reviewed by experts on the website, weighed, and a choice is then made on the final number.

The market values for players on Transfermarkt from its crowd-based approach are used by soccer scouts and clubs around the world. We will compare predictions generated by our models with Transfermarkt's values (using it as a source of truth) to determine our model's accuracy.

Transfermarkt had no API for us to attain the required data so we wrote code to scrape from the site from scratch using the BeautifulSoup library. Even though this was a time-consuming process, being in charge of our dataset allowed us to choose the features for our model and make feature engineering easier. For example, we chose not to include goalkeepers in our dataset as performance metrics for goalkeepers differ from outfield players. The different features from goalkeepers and different market values would potentially create a lot of noise in our dataset.

The scraped data required significant cleaning. For example, we had to convert values given in strings like "$50.00m" and "$700th." to float values and drop records for players who had numerous missing values on Transfermarkt. More details about data cleansing is in the scraping notebook provided.

After cleaning up the data, we ended up with 12,004 records of players. The features each player had were both characteristic and performance-based:

Characteristics

- **ID:** Transfermarkt ID.
    - A player's Transfermarkt ID, found in the URL for identification purposes.
- **Name:** player's name.
    - The full name of the player
- **Age:** player's age.
    - Reflects the player's experience and potential. Older players are thought to be less valuable investments since they are near the end of their careers.
- **Height**: player's height.
    - Large heights could indicate good heading ability that may increase the probability of scoring or preventing a goal.
- **Footedness**: whether the player is right or left-footed.
    - Footedness is important for set-pieces and tactical reasons (creating shooting/passing angles. Two-footedness is an advantageous footballing ability that also reflects players' flexibility.
- **Nationality**: player's country of birth or national team choice.
    - Some nationalities draw higher transfer fees due to the prestige of the nation's top league and/or national team.
- **League:** country and rank of the league the player plays in.
    - Similar to nationality, some leagues draw higher transfer fees due to the prestige of the league.
- **Contract Expiration:** the date the player's contract at their club expires in days.
    - Players with long contracts are more expensive than players at the end of their contracts as the selling club is usually worried about losing the player for free if their contract expires without selling them on to another club.
- **Position:** player's primary position.
    - The primary position of where a player plays on the field. This can be an important identifier for scouts and analysts as it can influence how they perceive other attributes. For example, height can be an important attribute for a Center-Back as they are usually defending set-pieces and balls in the air.

For each performance-based attribute, we had features for their all-time career statistics as well as the last season (19/20) and the current one being played now (20/21)

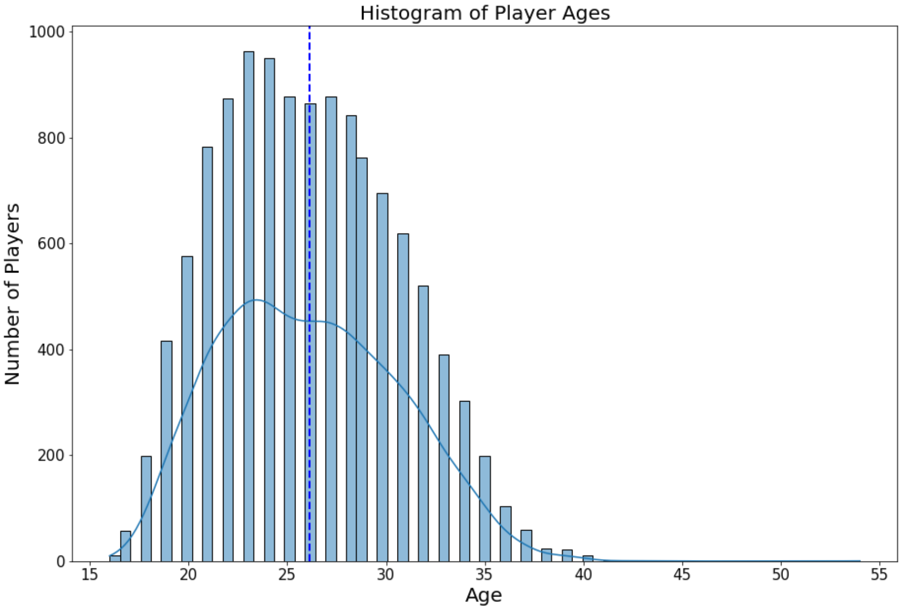
Performance

- **Games played**: number of games a player has played.
    - Shows if a player has been playing regularly for their team or is consistently on the bench or in the reserves.
- **Goals:** number of goals a player has scored.
    - Depending on the position, players with higher goal tallies are usually more valuable as goals win soccer matches.
- **Assists:** the number of assists a player has provided.
    - Depending on the position, players with higher assist tallies are usually more valuable as assists are usually necessary for goals to win soccer matches.

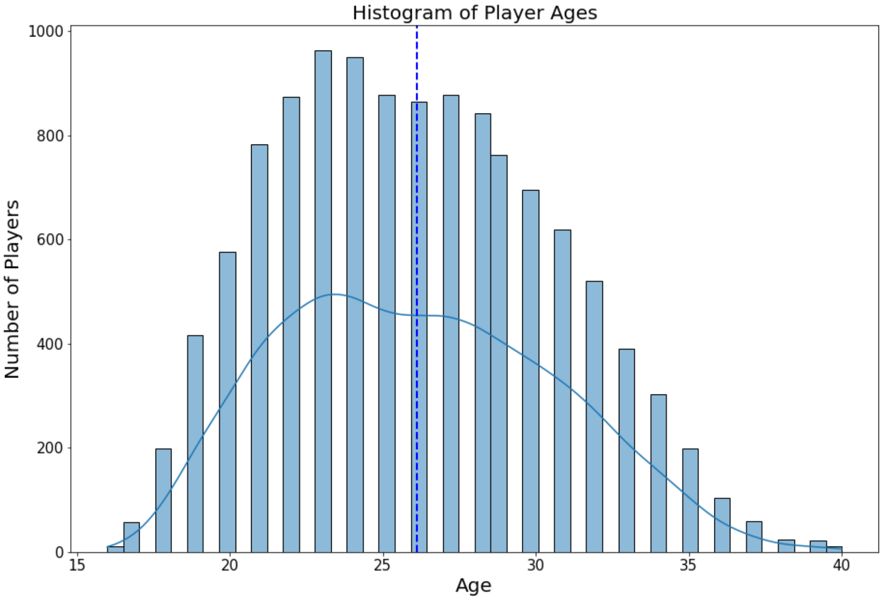
- **Cards:** number of yellow, yellow/red, and red cards received by a player.
    - Players with poor disciplinary records could be less valuable as these players can miss many games through suspension.

Data Exploration

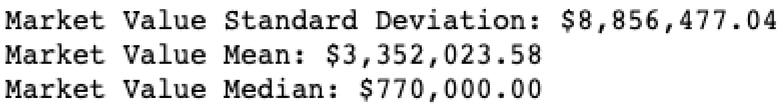
Data Exploration began with viewing a histogram of player ages. The data appeared to follow a normal distribution, however, there were a few outliers over the age of 40 that were removed.



Players at this age are not valued highly and are usually retiring within a season or two. A total of 4 players were removed who were over the age of 40.

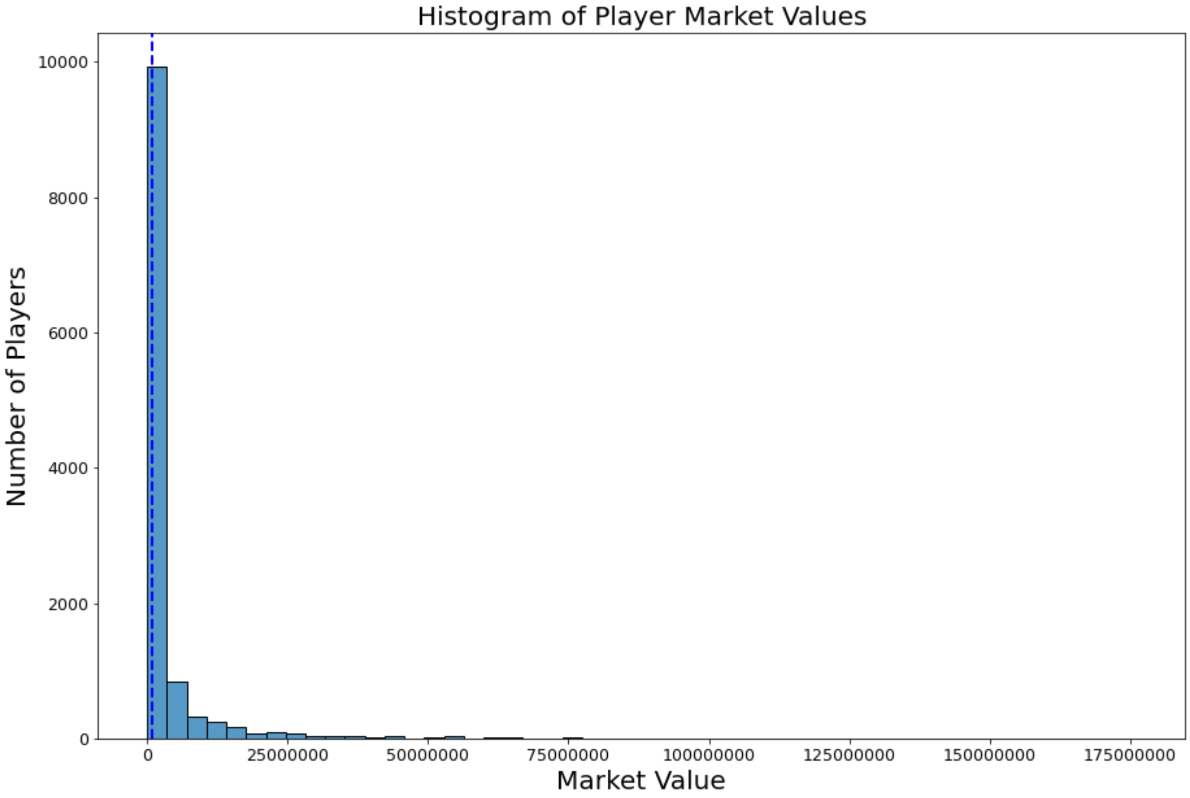


The mean, median, and standard deviation for Market Value can be seen below. A mean of $3.3m and a median of $770k, tells us that a majority of the market values are quite low, but a standard deviation of $8.8m alludes to some large values within the dataset.



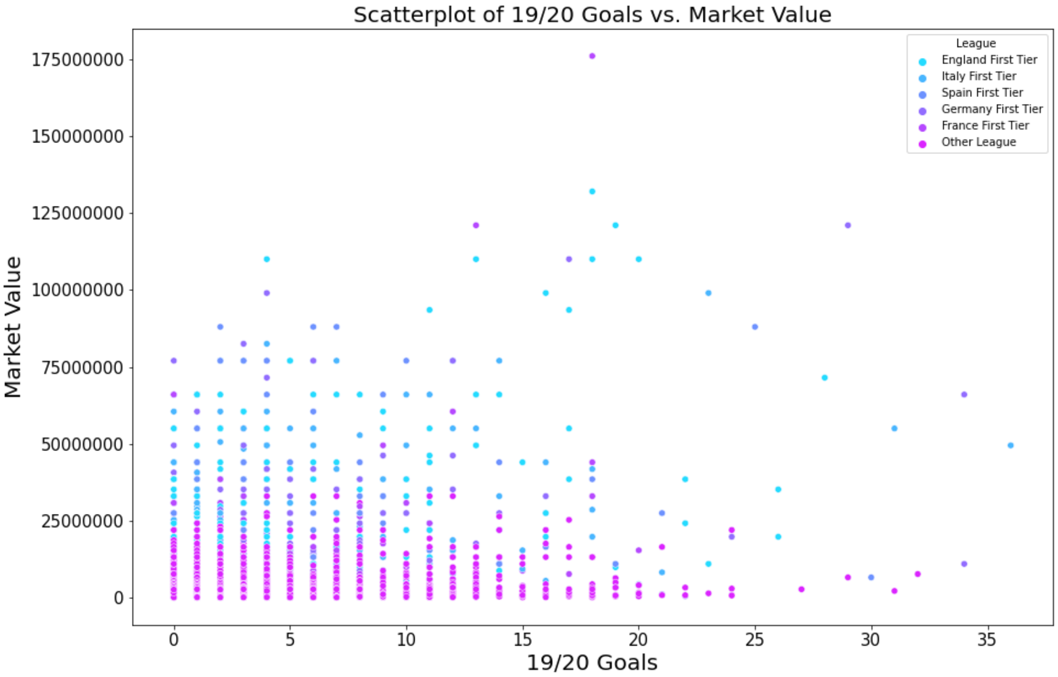
```
Market Value Standard Deviation: $8,856,477.04
Market Value Mean: $3,352,023.58
Market Value Median: $770,000.00
```

This can be better understood in the histogram of market values below. As previously mentioned, a majority of the market values fall below $1m as can be seen by the dashed line which represents the median. However, there are a few significantly higher market values for players as can be seen by the scale of the histogram extended to $175m.

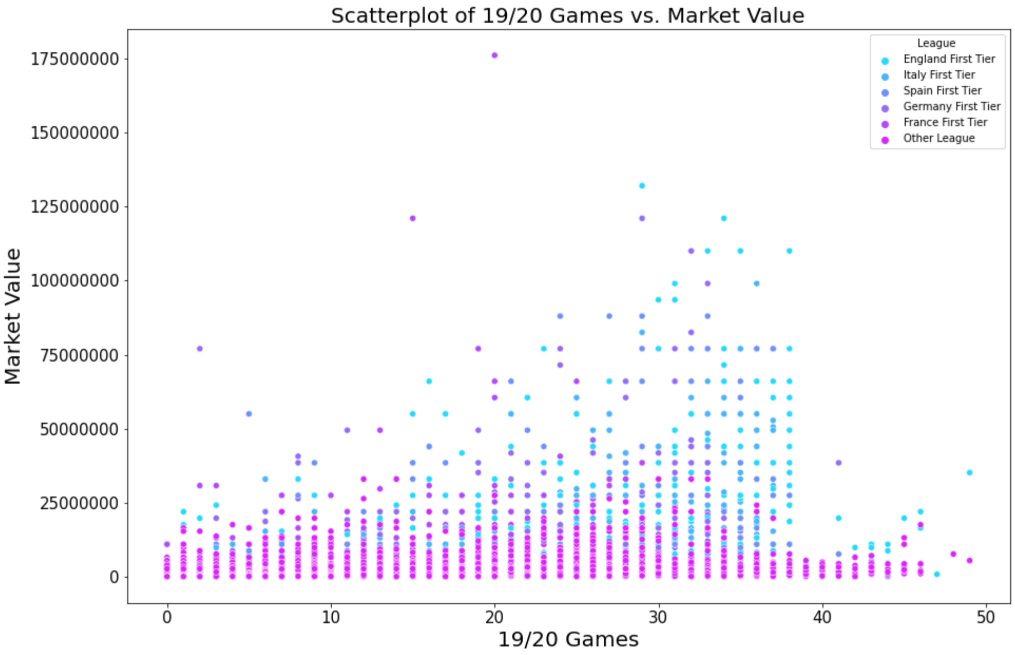


Soccer matches are won by scoring goals, so intuition would suggest that players who score more goals in a prestigious league, should be valued higher than others. The goal tallies of players from last season were compared against their market value and their league in the scatterplot below. It is evident that players who play in the “top 5” leagues (England First Tier, Spain First Tier, Italy First Tier, Germany First Tier, and France First Tier) and who score more goals, tend to be worth more than players who play and score goals in other, less prestigious leagues.

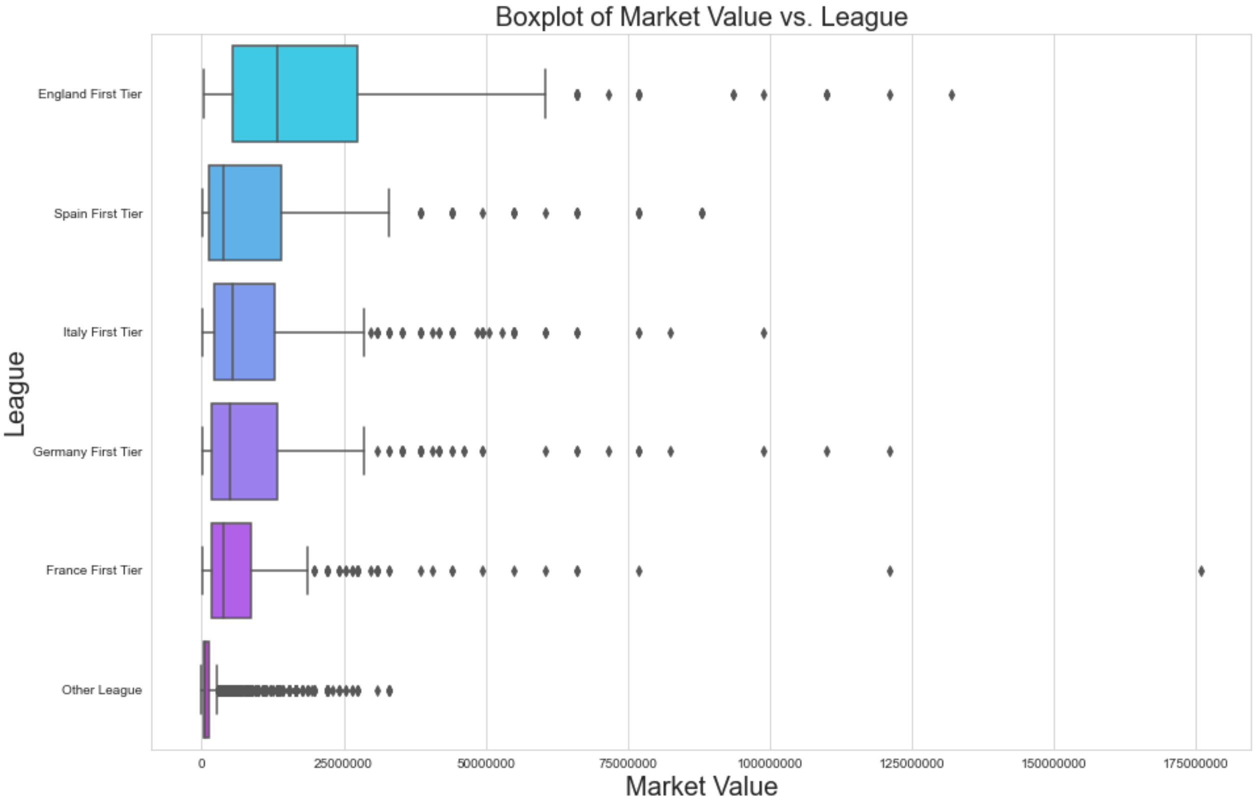
The “top 5” leagues were determined by viewership, domain knowledge, and FiveThirtyEight’s SPI ranking for teams in which most top teams in the world fell within one of these leagues [6].



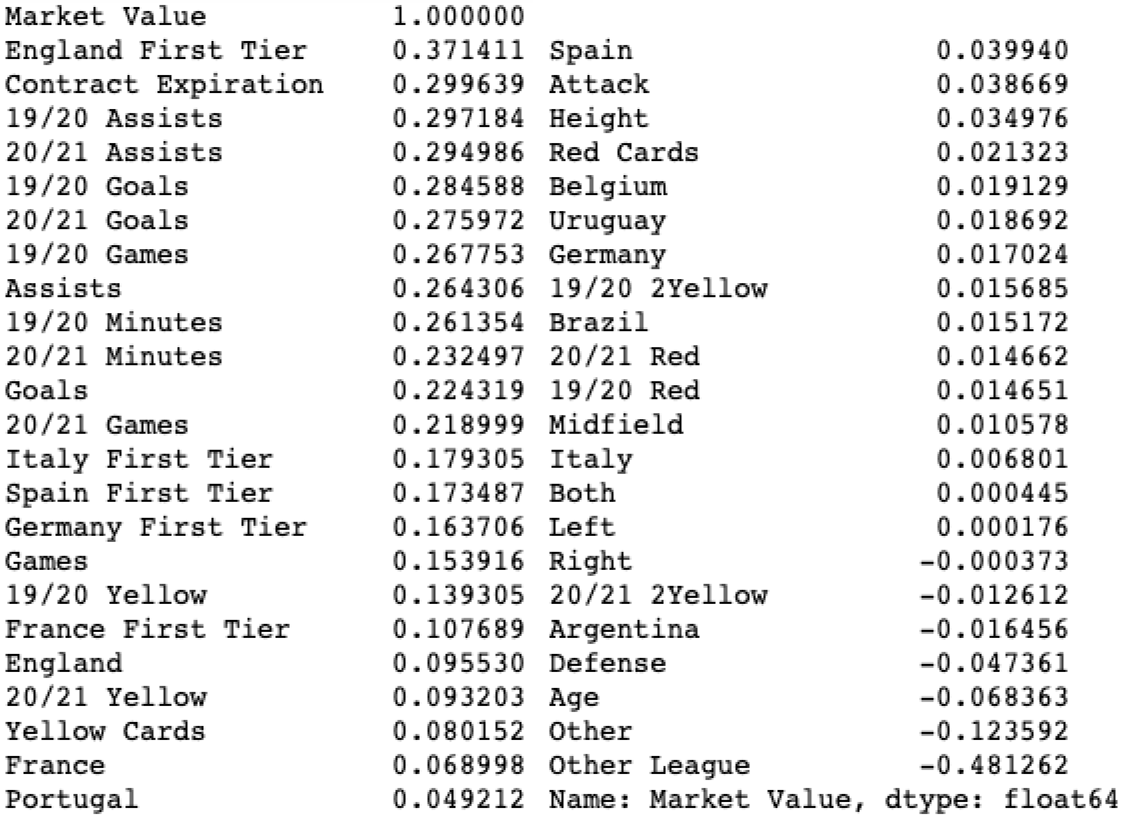
Not all players contribute heavily to goals but can still be valued highly such as defenders or defensive midfielders. A scatterplot showing the number of games played last season versus market value provided a visual insight into the number of games players played last season, in which league, and how they were valued. As expected, players who played a lot of games in the top leagues tended to be valued higher than players who played in another league.



Similar to the histogram of market values, a boxplot of market values separated by League was used to further understand the distribution of market values. It can be seen in the boxplot below that the players in the “top 5” leagues are valued significantly higher than players in other leagues which is consistent with our knowledge so far.

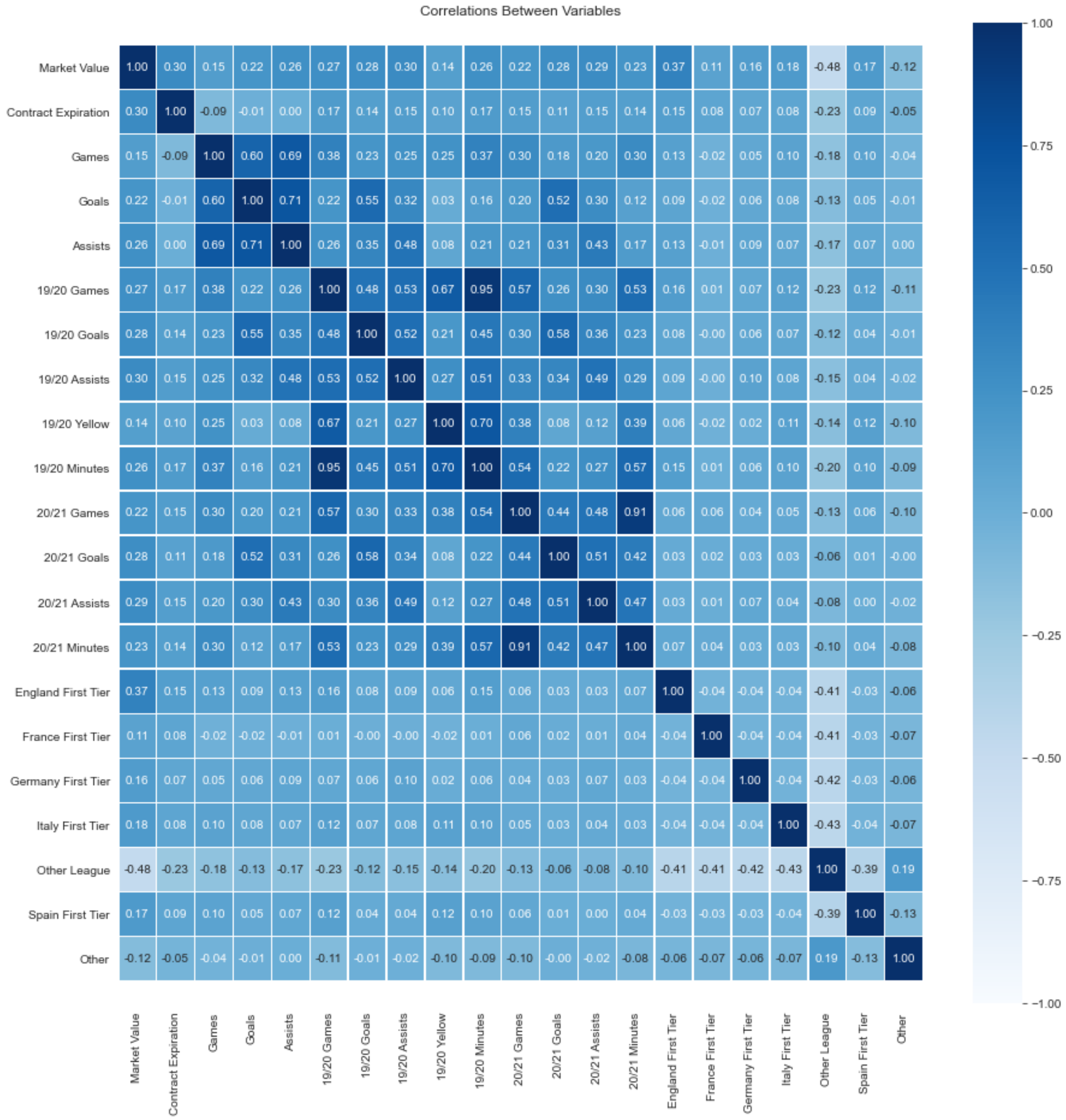


Linear correlation is a useful method to discover any linear correlation between features and response variables and between features themselves. We observed the correlation between the features and response variable and removed features that had a correlation value between -0.1 and 0.1 for visualization purposes and because these are weak relationships.



```
Market Value           1.000000
England First Tier     0.371411 Spain                 0.039940
Contract Expiration    0.299639 Attack                0.038669
19/20 Assists          0.297184 Height                0.034976
20/21 Assists          0.294986 Red Cards             0.021323
19/20 Goals            0.284588 Belgium               0.019129
20/21 Goals            0.275972 Uruguay               0.018692
19/20 Games            0.267753 Germany               0.017024
Assists                0.264306 19/20 2Yellow         0.015685
19/20 Minutes          0.261354 Brazil                0.015172
20/21 Minutes          0.232497 20/21 Red             0.014662
Goals                  0.224319 19/20 Red             0.014651
20/21 Games            0.218999 Midfield              0.010578
Italy First Tier       0.179305 Italy                 0.006801
Spain First Tier       0.173487 Both                  0.000445
Germany First Tier     0.163706 Left                  0.000176
Games                  0.153916 Right                -0.000373
19/20 Yellow           0.139305 20/21 2Yellow        -0.012612
France First Tier      0.107689 Argentina            -0.016456
England                0.095530 Defense              -0.047361
20/21 Yellow           0.093203 Age                  -0.068363
Yellow Cards           0.080152 Other                -0.123592
France                 0.068998 Other League         -0.481262
Portugal               0.049212 Name: Market Value, dtype: float64
```

The final correlation DataFrame was used to plot the below heatmap. As can be seen in the output above as well as the heatmap, Other League, England First Tier, and Contract Expiration have medium strength correlations with Market Value. In this exploration, we were also looking for strong relationships between variables as this would create issues of multicollinearity.



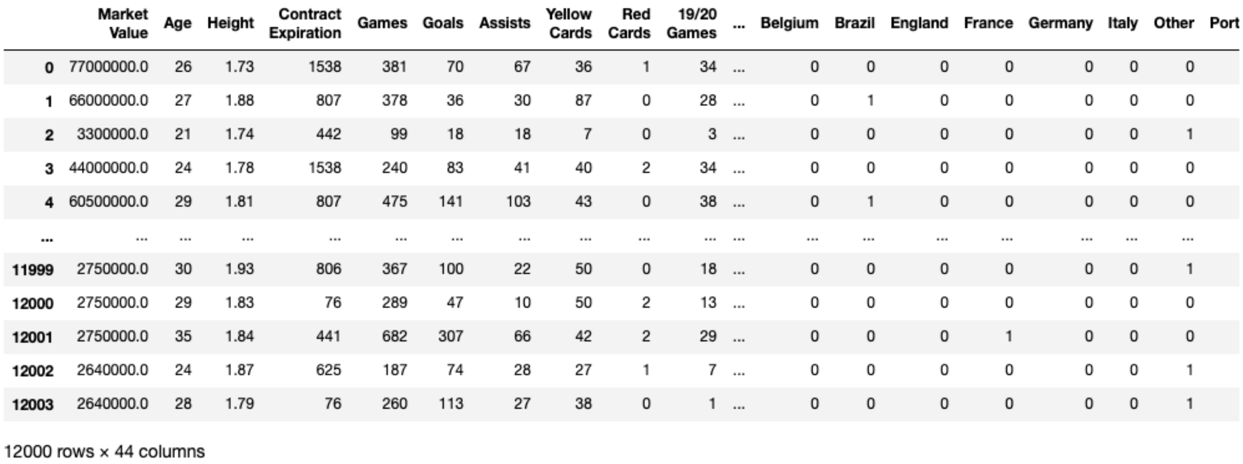
The only two strong correlations between variables were:

- 19/20 Games and 19/20 Minutes (0.95)
- 20/21 Games and 20/21 Minutes (0.91)

We decided to remove 19/20 Minutes and 20/21 Games as they had the weakest correlation to the response variable.

Approach and Methodology

The dataset has several categorical variables such as nationality and league. We used one-hot encoding to include them in our model. However, since our dataset included over 20 leagues and 100 nationalities, one hot encoding each category would create an excessive amount of features. As a result, we only encoded the top 5 leagues and nationalities (determined by the official UEFA coefficient ranking and FIFA world rankings respectively). If a player plays in a league or is from a nationality not in the "top", they would be categorized in the "other" league/nationality.



	Market Value	Age	Height	Contract Expiration	Games	Goals	Assists	Yellow Cards	Red Cards	19/20 Games	...	Belgium	Brazil	England	France	Germany	Italy	Other	Port
0	77000000.0	26	1.73	1538	381	70	67	36	1	34	...	0	0	0	0	0	0	0	
1	66000000.0	27	1.88	807	378	36	30	87	0	28	...	0	1	0	0	0	0	0	
2	3300000.0	21	1.74	442	99	18	18	7	0	3	...	0	0	0	0	0	0	1	
3	44000000.0	24	1.78	1538	240	83	41	40	2	34	...	0	0	0	0	0	0	0	
4	60500000.0	29	1.81	807	475	141	103	43	0	38	...	0	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
11999	2750000.0	30	1.93	806	367	100	22	50	0	18	...	0	0	0	0	0	0	1	
12000	2750000.0	29	1.83	76	289	47	10	50	2	13	...	0	0	0	0	0	0	0	
12001	2750000.0	35	1.84	441	682	307	66	42	2	29	...	0	0	0	1	0	0	0	
12002	2640000.0	24	1.87	625	187	74	28	27	1	7	...	0	0	0	0	0	0	1	
12003	2640000.0	28	1.79	76	260	113	27	38	0	1	...	0	0	0	0	0	0	1	

12000 rows × 44 columns

The final DataFrame after Data Cleaning and Data Exploration had 12,000 rows and 44 columns. We realized that we still needed to perform some feature selection based on the feature importance of each model to reduce the dimensionality of our dataset.

Next, we randomly split the dataset into training and testing sets using a 75/25 split. Finally, we used the StandardScaler to standardize our training, testing, and validation sets (mean of 0 and standard deviation of 1 for each feature) to use for linear regression.

We started by training initial models and used them as baselines to improve upon. We will primarily use recursive feature elimination cross-validation (RFECV) and GridSearchCV to do cross-validation, feature selection, and hyperparameter tuning. These scikit-learn packages have k-fold cross validation built into them and they select features/tune hyperparameters based on how well they perform on validation sets.

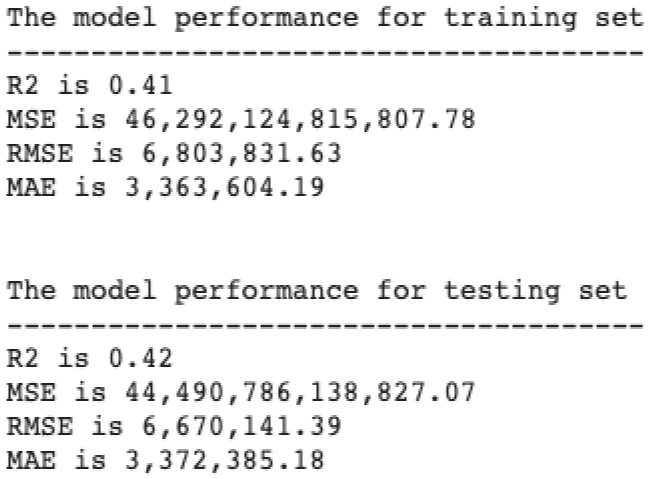
Machine Learning Models

We used the following machine learning models on our dataset:

- Linear Regression
- Decision Tree Regressor
- AdaBoost Regressor
- Random Forest Regressor
- Gradient Boosting Regressor

Linear Regression

Linear regression is a simple regression model that we use as a baseline compared to other models. It does not require any hyperparameter tuning.

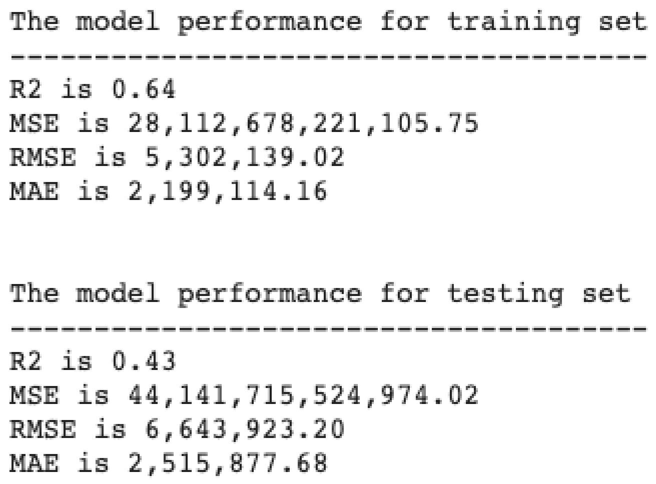


```
The model performance for training set
--------------------------------------
R2 is 0.41
MSE is 46,292,124,815,807.78
RMSE is 6,803,831.63
MAE is 3,363,604.19


The model performance for testing set
--------------------------------------
R2 is 0.42
MSE is 44,490,786,138,827.07
RMSE is 6,670,141.39
MAE is 3,372,385.18
```

The results from linear regression were not great, but the 0.41 R^2 does signify there is a relationship and correlation between our features and target value. Even though the mean square error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) were very high, the linear regression model did a great job of not overfitting to the training set.

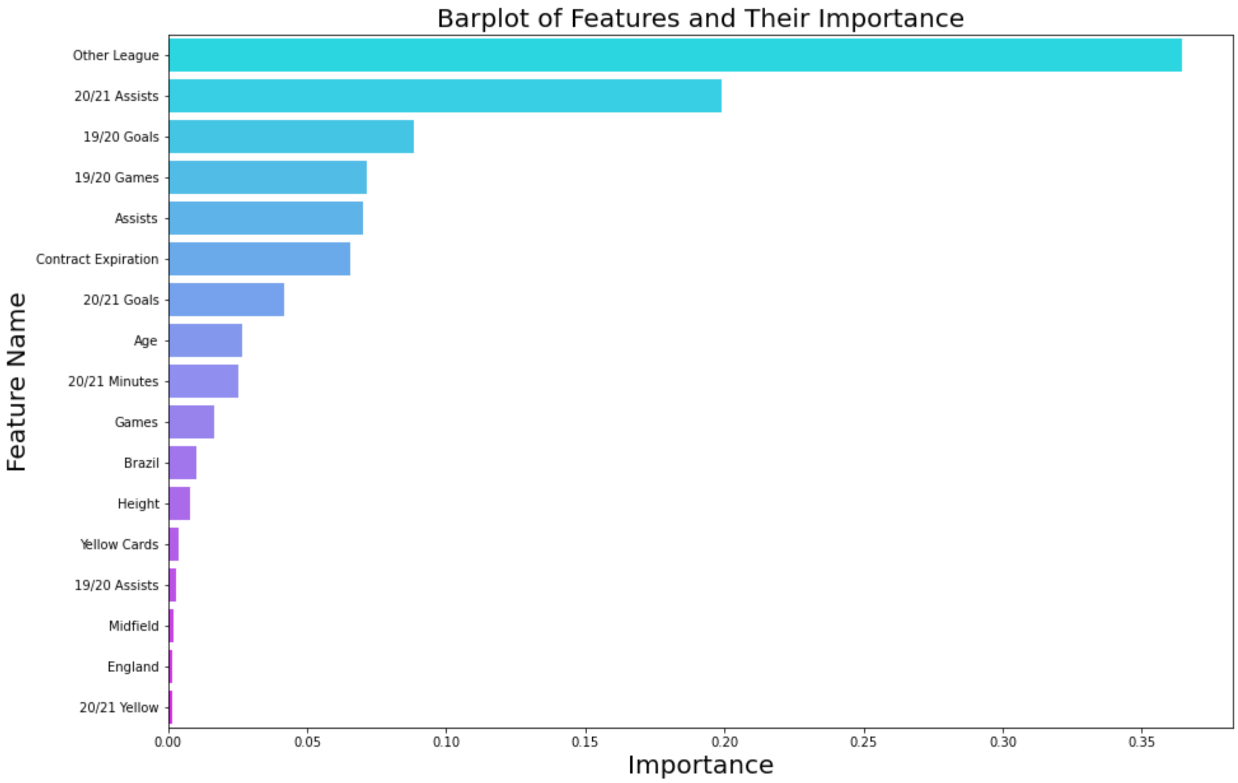
Decision Tree

Baseline Decision Tree
For our decision tree, we manually validated the optimal maximum tree depth. The optimal maximum depth was 6. Even after getting the optimal max tree depth, our decision tree model still showed some overfitting.



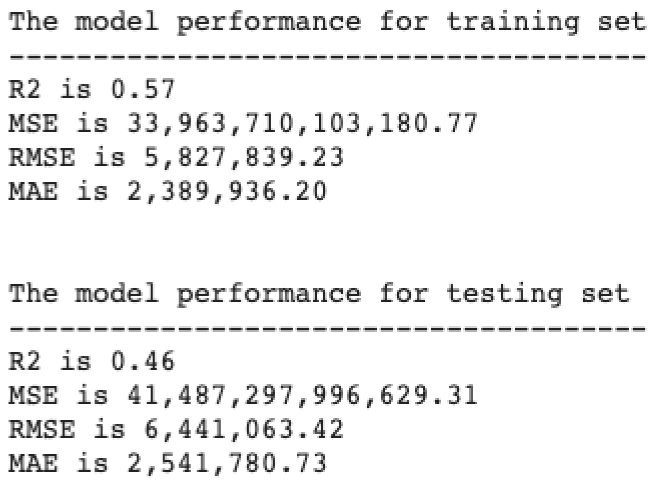
```
The model performance for training set
--------------------------------------
R2 is 0.64
MSE is 28,112,678,221,105.75
RMSE is 5,302,139.02
MAE is 2,199,114.16


The model performance for testing set
--------------------------------------
R2 is 0.43
MSE is 44,141,715,524,974.02
RMSE is 6,643,923.20
MAE is 2,515,877.68
```

The decision tree model found Other League (playing in a league outside the top 5), Contract Expiration, 20/21 assists, and 19/20 games and goals as the most important features.



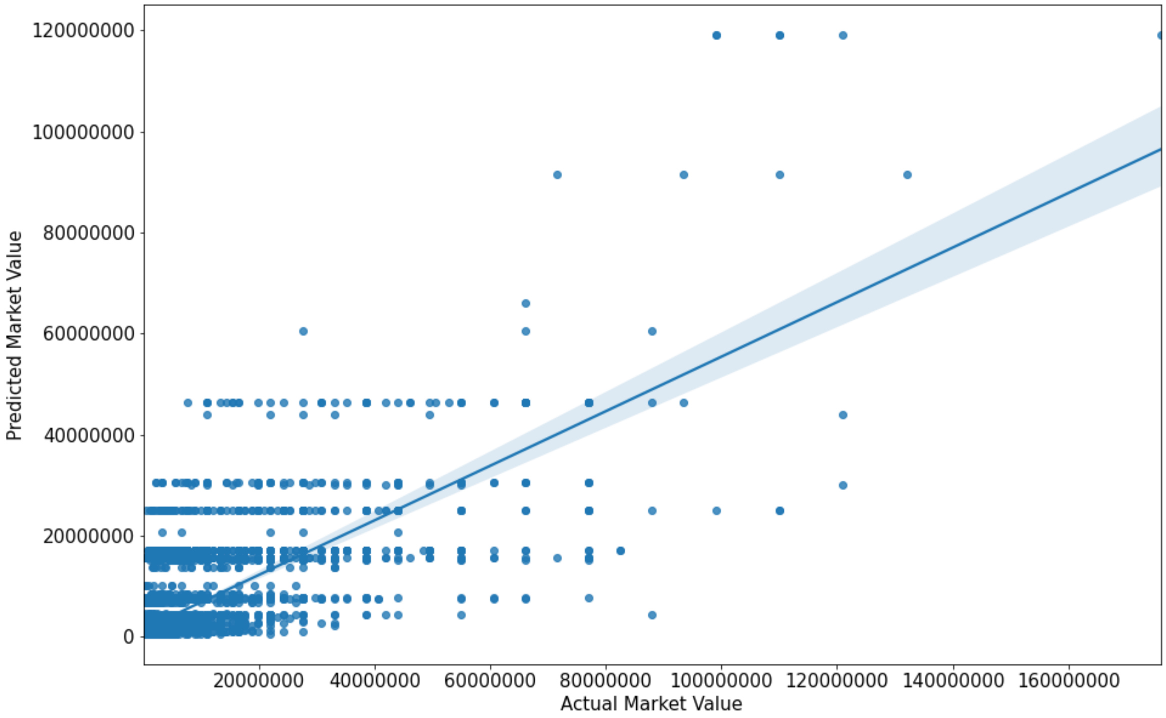
Optimized Decision Tree

We then did some manual feature selection to help avoid some of this overfitting. After plotting the feature importances (or variable importance) above, we noticed that the model only found about 20 features of significant importance. A majority of the features in our dataset did not contribute to the prediction. So, we ran the model again with the 20 most important features.



```
The model performance for training set
--------------------------------------
R2 is 0.57
MSE is 33,963,710,103,180.77
RMSE is 5,827,839.23
MAE is 2,389,936.20


The model performance for testing set
--------------------------------------
R2 is 0.46
MSE is 41,487,297,996,629.31
RMSE is 6,441,063.42
MAE is 2,541,780.73
```

Below is a visual representation of the actual versus the predicted values for the final decision tree model.

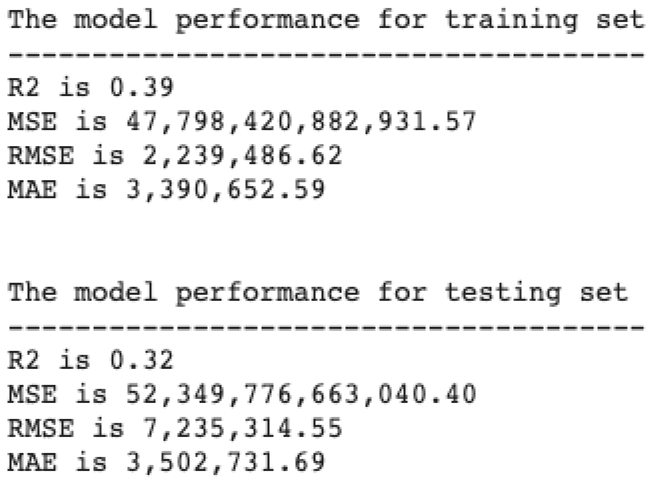


Feature selection reduced the overfitting of the model slightly, by improving the R^2 on the testing set while decreasing the R^2 on the training set. Since we had already done manual hyperparameter tuning and because the decision tree model does not have as many hyperparameters as the other models, we did not perform any further cross-validation.

AdaBoost Regression

Baseline AdaBoostRegressor

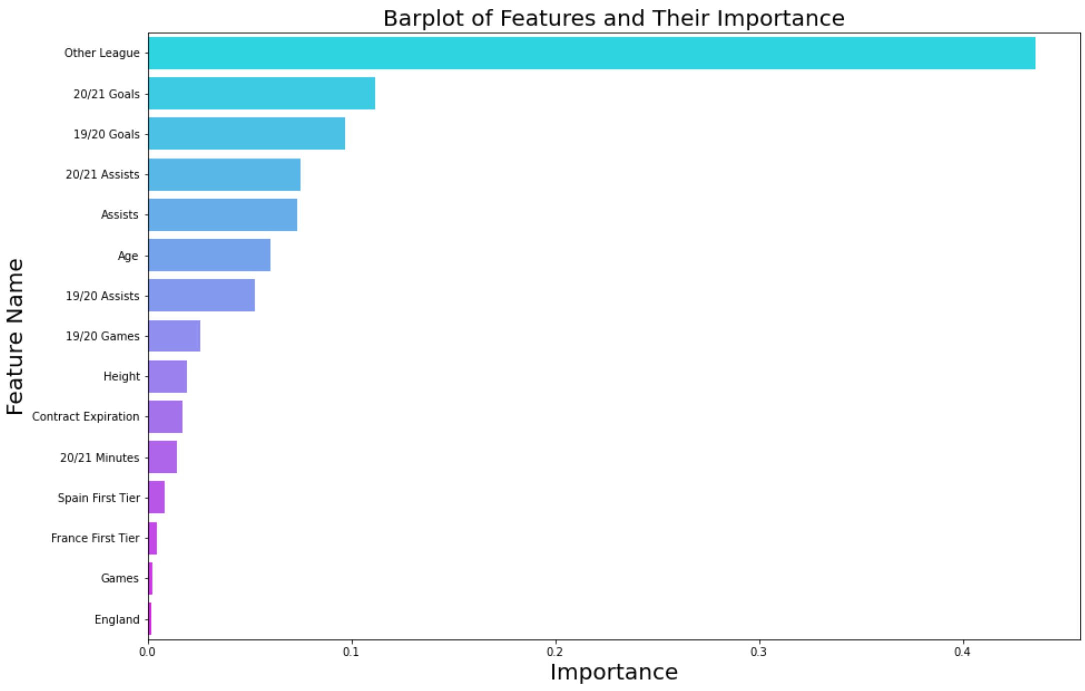
We used an AdaBoost regressor, with an alpha of 0.1 and 75 estimators, to get baseline results for the model. We used a decision tree regressor as our base estimator as it is the most common and effective weak learner.



```
The model performance for training set
--------------------------------------
R2 is 0.39
MSE is 47,798,420,882,931.57
RMSE is 2,239,486.62
MAE is 3,390,652.59


The model performance for testing set
--------------------------------------
R2 is 0.32
MSE is 52,349,776,663,040.40
RMSE is 7,235,314.55
MAE is 3,502,731.69
```

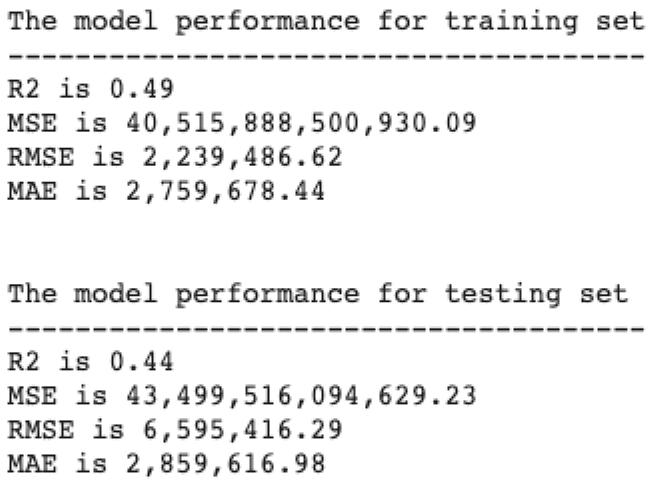
Our initial results were not great. The baseline model performed slightly worse than linear regression.

The AdaBoost regressor found mostly the same features important as the other models.

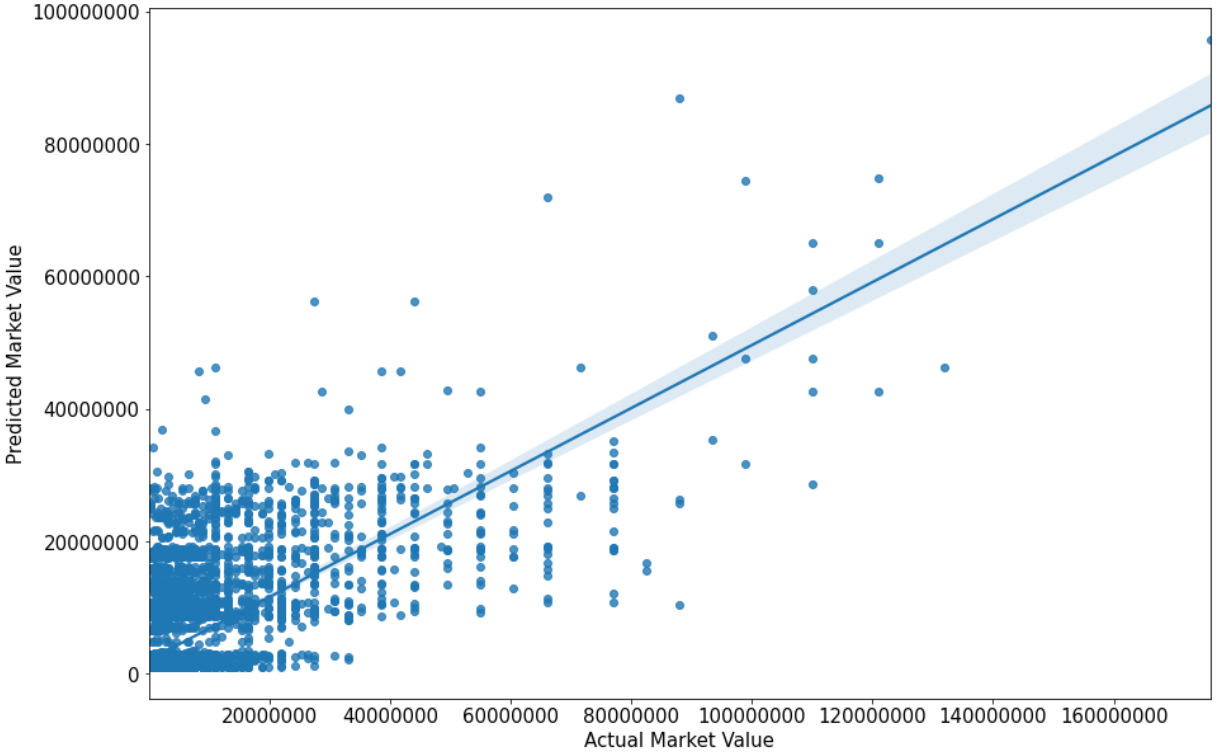


Optimized AdaBoostRegressor

To optimize our baseline model, we created a pipeline that first used recursive feature elimination cross-validation (RFECV) to select the most important features, and then used GridSearchCV to get the optimal hyperparameters. Since AdaBoost regressor has a tradeoff between the learning rate and the number of estimators, this grid search was essential to improve our results. The grid search found an optimal learning rate of 0.03 and 75 for the number of estimators.



```
The model performance for training set
--------------------------------------
R2 is 0.49
MSE is 40,515,888,500,930.09
RMSE is 2,239,486.62
MAE is 2,759,678.44


The model performance for testing set
--------------------------------------
R2 is 0.44
MSE is 43,499,516,094,629.23
RMSE is 6,595,416.29
MAE is 2,859,616.98
```

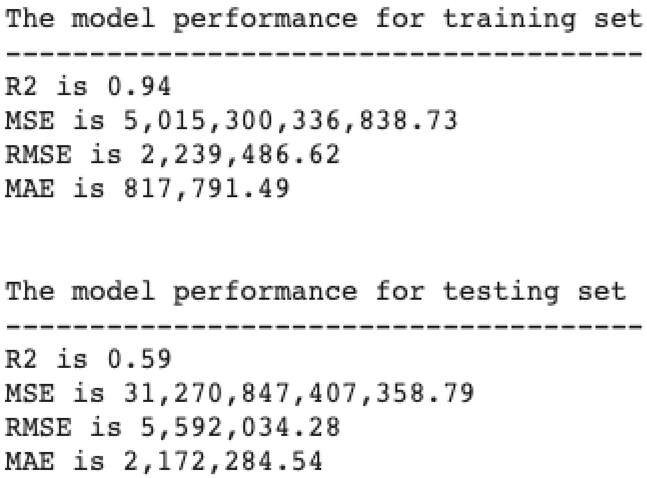


Our model performed much better after completing these transformations, but the results were still not as impressive as we had hoped for. Similar to linear regression, the AdaBoost regressor did a good job of not overfitting.

Random Forest

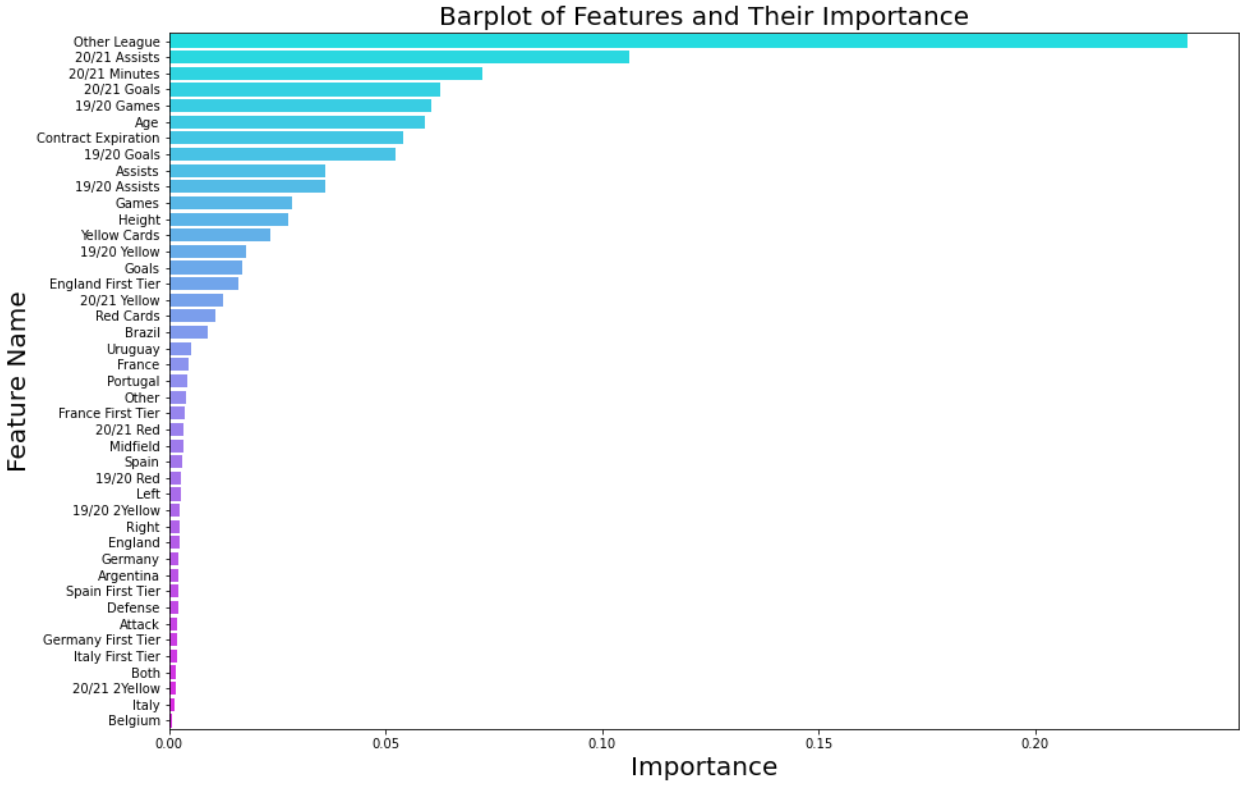
Baseline Random Forest Regression

The Random Forest regressor performed much better on the testing set than the previous models but showed some strong overfitting as it had a very high R^2 value on the training set.



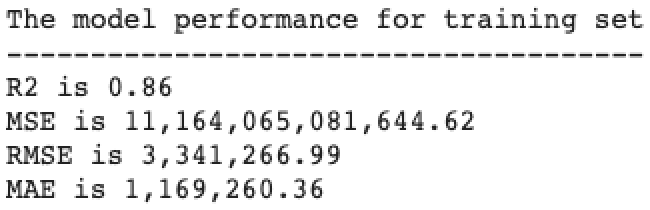
```
The model performance for training set
--------------------------------------
R2 is 0.94
MSE is 5,015,300,336,838.73
RMSE is 2,239,486.62
MAE is 817,791.49


The model performance for testing set
--------------------------------------
R2 is 0.59
MSE is 31,270,847,407,358.79
RMSE is 5,592,034.28
MAE is 2,172,284.54
```

The Random Forest regressor found mostly the same features important as the other models but found additional slight importances in all the features.

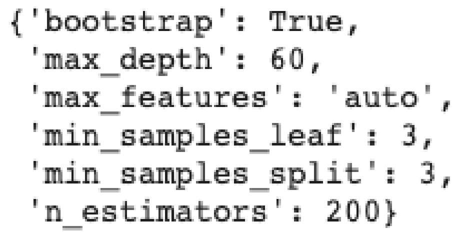


Optimized Random Forest Regression

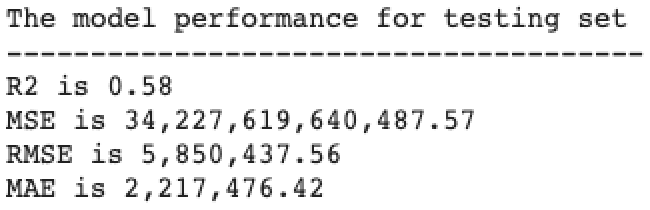
To optimize the model and reduce overfitting, we created a similar type of pipeline like we did for AdaBoost. Random Forest has a large number of parameters, so we implemented a RandomizedSearchCV to get a sense of the range of values to try. After running the randomized search, the data was trained on a pipeline that sequentially did feature selection and hyperparameter tuning. The optimal parameters found and the results are displayed below.



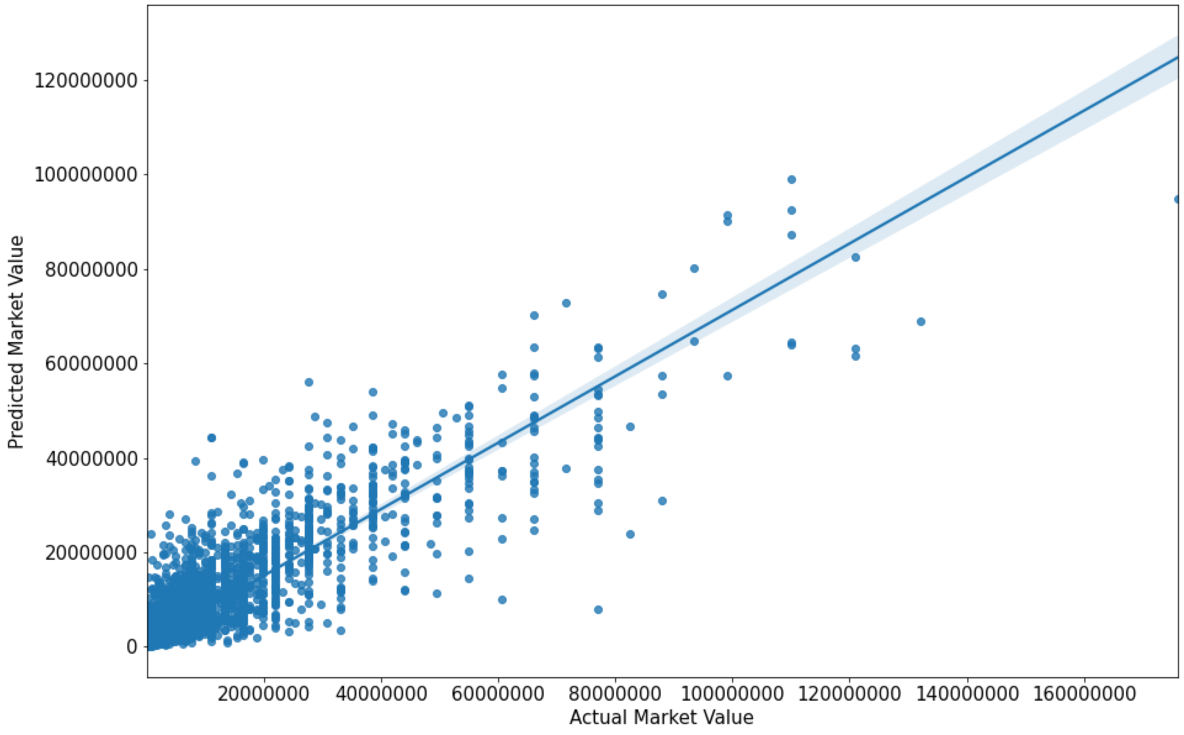
```
The model performance for training set
--------------------------------------
R2 is 0.86
MSE is 11,164,065,081,644.62
RMSE is 3,341,266.99
MAE is 1,169,260.36
```



```
{'bootstrap': True,
 'max_depth': 60,
 'max_features': 'auto',
 'min_samples_leaf': 3,
 'min_samples_split': 3,
 'n_estimators': 200}
```



```
The model performance for testing set
--------------------------------------
R2 is 0.58
MSE is 34,227,619,640,487.57
RMSE is 5,850,437.56
MAE is 2,217,476.42
```

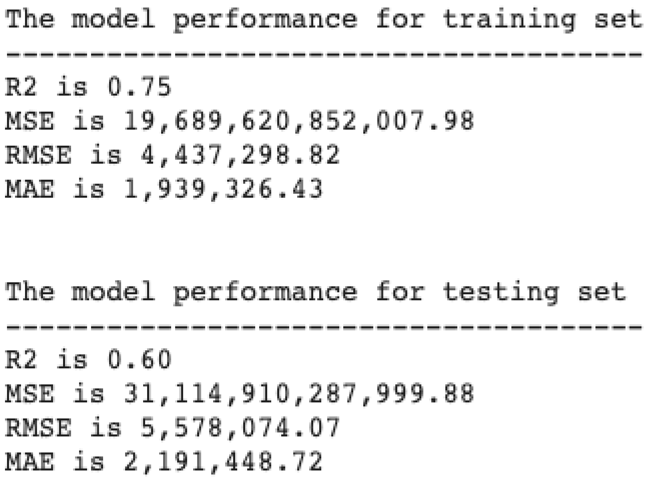


The optimized model improved overfitting slightly by reducing the training R^2, but unfortunately, it did not improve the testing set R^2. Nonetheless, an R^2value of about 0.6 on the testing set is the highest of any model, and this model is doing a much better job at predicting market value.

Gradient Boosting

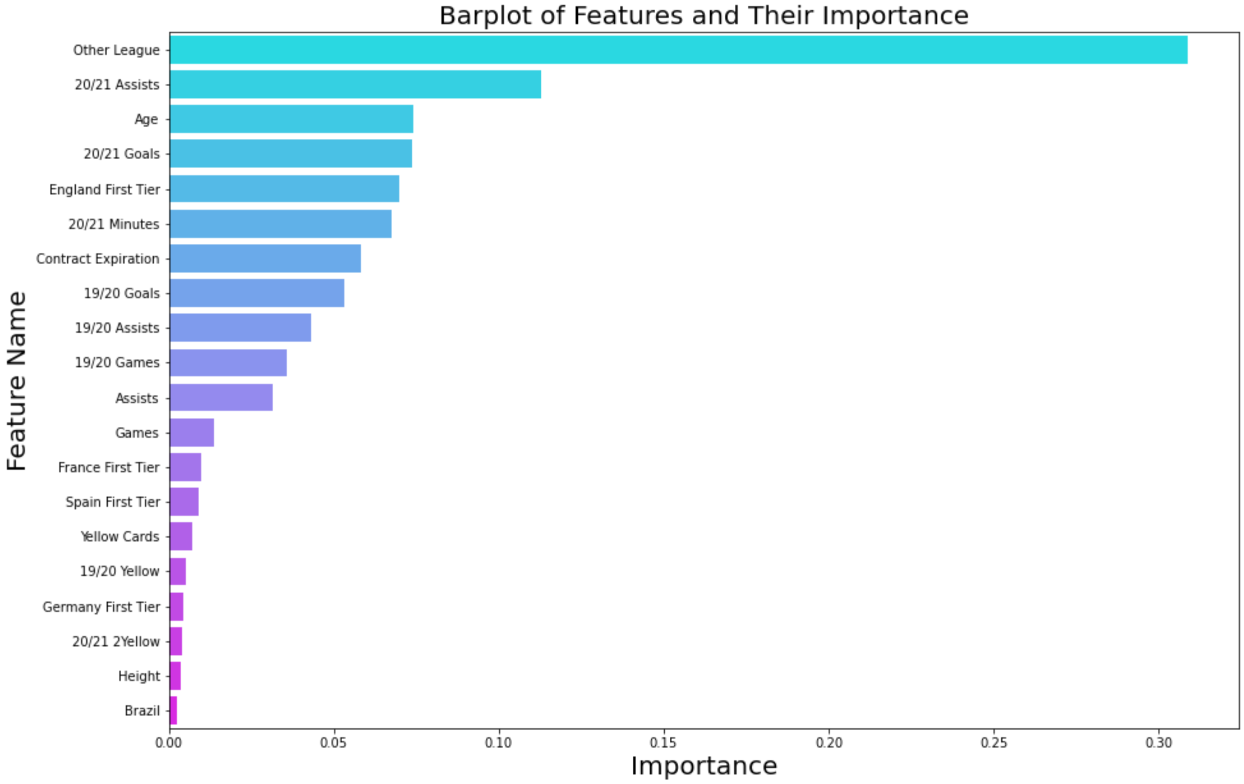
Baseline Gradient Boosting Regressor

The Gradient Boosting model, which also uses a decision tree as the base model, performed much better compared to the other models’ baseline.



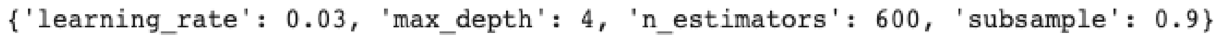
```
The model performance for training set
--------------------------------------
R2 is 0.75
MSE is 19,689,620,852,007.98
RMSE is 4,437,298.82
MAE is 1,939,326.43


The model performance for testing set
--------------------------------------
R2 is 0.60
MSE is 31,114,910,287,999.88
RMSE is 5,578,074.07
MAE is 2,191,448.72
```

The model found the below features important, giving Other League the most importance.

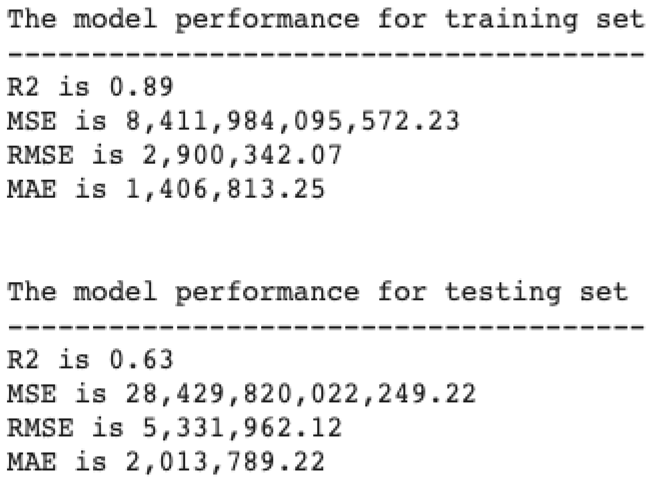


Optimized Gradient Boosting Regressor

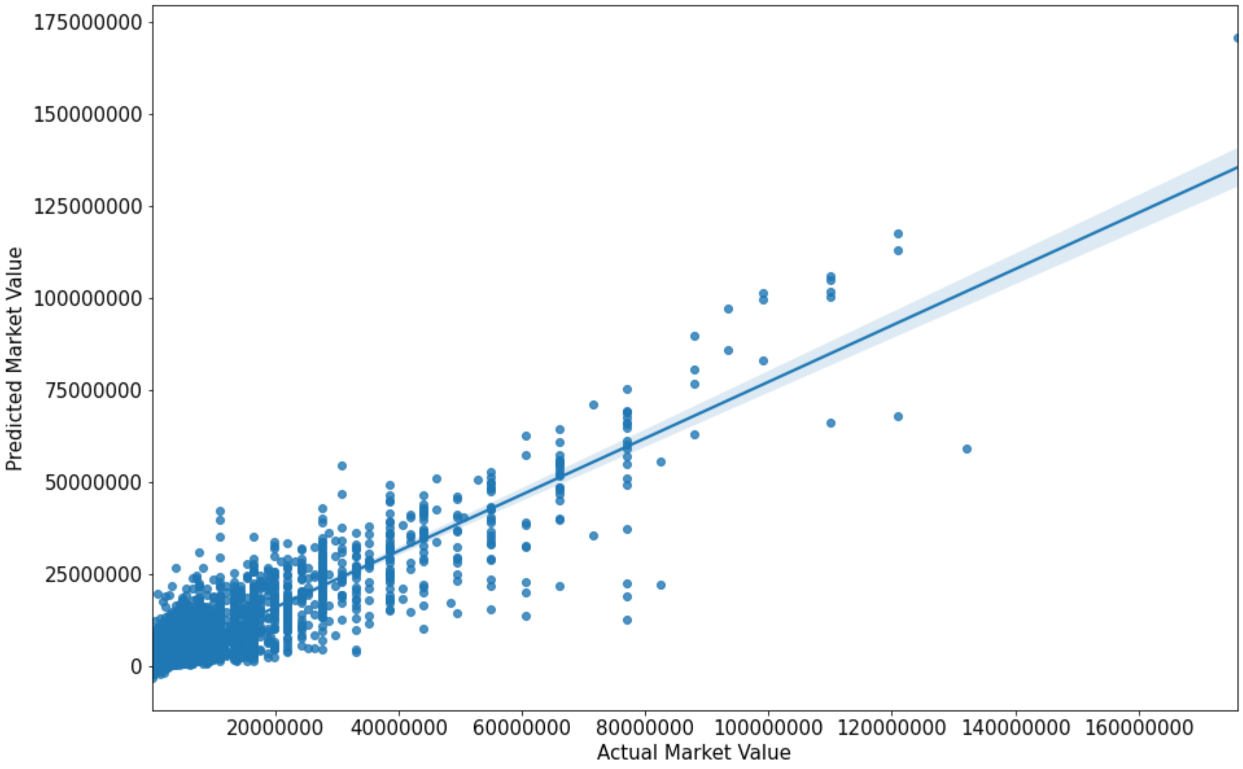
Similar to AdaBoost and Random Forest, we created a pipeline that uses RFECV to do feature selection and then grid search to find the most optimal hyperparameters. The optimal hyperparameters were:



```
{'learning_rate': 0.03, 'max_depth': 4, 'n_estimators': 600, 'subsample': 0.9}
```



```
The model performance for training set
--------------------------------------
R2 is 0.89
MSE is 8,411,984,095,572.23
RMSE is 2,900,342.07
MAE is 1,406,813.25


The model performance for testing set
--------------------------------------
R2 is 0.63
MSE is 28,429,820,022,249.22
RMSE is 5,331,962.12
MAE is 2,013,789.22
```



After the optimization, the model scored higher on both the training and testing set. While it did not quite improve the overfitting problem as much, it still, generally, improved the model's performance. The Gradient Boosting regressor has proven to be the best model we have tested.

Conclusion

All of our models were able to get an R^2 value greater than 0.4 on unseen data, with Random Forest Regression and Gradient Boosting Regression achieving an R^2 value greater than and around 0.6. Even though each model had a high mean squared error, they can sensibly predict the market value of soccer players. We did counteract some overfitting through feature selection and hyperparameter tuning, but Random Forest, Gradient Boosting, and Decision Tree still somewhat overfit the training data. The features our model found the most important were Other League (playing in a league outside the “top 5”), England First Tier (playing in the Premier League in England), goals and assists (for the 20/21 and 19/20 seasons), Contract Expiration, and Age. All of these features being important to market value is very reasonable and encouraging for our model’s validity

To reduce overfitting and improve model performance, we would need to collect many more records. Having additional records could also allow us to retain the individuality between leagues and nations as opposed to combining them into an “Other” category. However, there are not many additional records within each league we could have collected as there are only so many players within a league and we used them all besides certain outliers and players with missing data. It might be useful to look at more detailed data on player performance such as metrics during a game that is not just goals, assists, etc. Football clubs and scouts spend a lot of money on more detailed performance data as most of this data is behind a paywall. That is why attaining that data was out of scope for this project but should be considered in the future.

Finally, we could try to change the response variable to be a transfer fee as opposed to the market value supplied by Transfermarkt. We could use machine learning to predict what the transfer fee will be for a player based on the features. However, this has many limitations as often clubs will overpay or underpay for a player as transfer fees can be determined by many factors specific to the club. In our opinion, it is better to predict the market value for a player to instruct clubs whether they are getting a good deal or not.

Team Member Contribution

The following is how the work was divided between team members:

Jalaj Singh

- Data Scraping/Data Cleaning
- Data Exploration
- Base models, cross-validation, grid-search (optimized models)
- Conclusion

Bennett Thorson

- Problem Statement
- Data Cleaning
- Data Exploration
- Base models and feature importance
- Conclusion

Code and Presentation Links

Presentation:
https://docs.google.com/presentation/d/11wQzCuihKtwfL3_8M00WEIxI-aOIvqz-rqB0pqGiEVM/edit?usp=sharing

Data Scraping Code:
https://drive.google.com/file/d/1bNkMhoNhUxr84a4uckFI3nHIdo5sHJYM/view?usp=sharing

Final Project Code:
https://colab.research.google.com/drive/13e7LnAxGG2tLAJV710XwvrEuuumbgOQ6?usp=sharing