Predicting NBA Player Contract Value Using Per 36 Player Statistics

Problem Statement

NBA organizations have been signing professional basketball players to increasingly lucrative deals in recent years and many have been disappointed by underperforming players. In other cases, the players themselves are forced to wonder why they are not being paid more when it seems like they are playing well and contributing to the team's success. Both the team and the players want to make a deal that helps them win and earn money.

How can NBA organizations and player agents predict player value so that a fair deal for both parties can be reached? What features drive value? Is there a way to accurately predict contract value based on prior performance?

Data Wrangling

Three sets of data were combined to form one dataset.

1. Player Per 36 Statistics - performance values adjusted for 36 minutes of game time
2. Player Contracts - contract values, type of contract, team, etc.
3. NBA Organization values - Forbes valuation of NBA organizations

The Player Per 36 Statistics and Player Contracts datasets were first merged on the Player column. The merged dataset contains only players that have both statistics and contract information available. Standard data cleaning was performed, including checking for nulls, dealing with inconsistent strings, removing repeated rows, etc. After that, the NBA organizational values were grouped into three categories (small, medium, or large) corresponding to the bottom 25%, middle 50%, and top 25% of all values. Both this information and the raw dollar amount was added to the dataset. Lastly, the data was sorted by contract value so the highest earning players appeared at the top of the list.

Once all data was merged, cleaned, and sorted, the final dataset contained 344 rows and 32 columns. This represents 344 individual players and 32 features. Since NBA teams are allowed to have 15 rostered players and there are 30 teams, a full dataset would come to at least 450 players, possibly more as some players are dropped mid season and others are signed from other leagues. For reference, in the 2019/20 season 529 players suited up for at least one game. This 344 player dataset represents approximately ⅔ of the NBA player population for the year.

Exploratory Data Analysis

Several aspects of the data needed to be investigated before determining how to approach the problem statement. My exploratory data analysis consisted of visualizing the relationships between different variables as well as visualizing the distribution of salaries among different groups of players.

The distribution across the five main positions was plotted to see what types of players are in the dataset (see Figure 1). Small Forwards were represented the least, with 56 players, whereas Shooting Guards were represented the most with 86 players. Though the count at each position is not equal, the dataset is not heavily unbalanced and each of the positions is represented by a decent number of players.

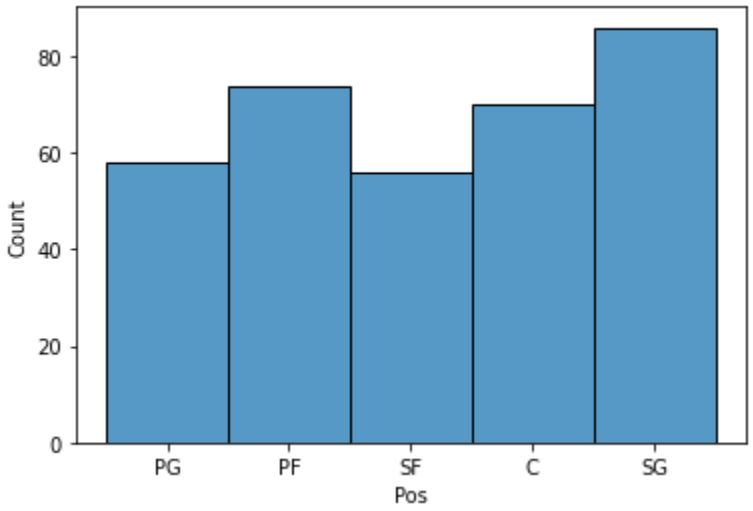

Figure 1: Number of players at each position.

The contract values differ by position, though there is not much variation in the median values relative to the overall range of contract values in the dataset. Comparing the contract value distribution for each position, it appears that the highest paid position is Point Guard (see Figure 2).

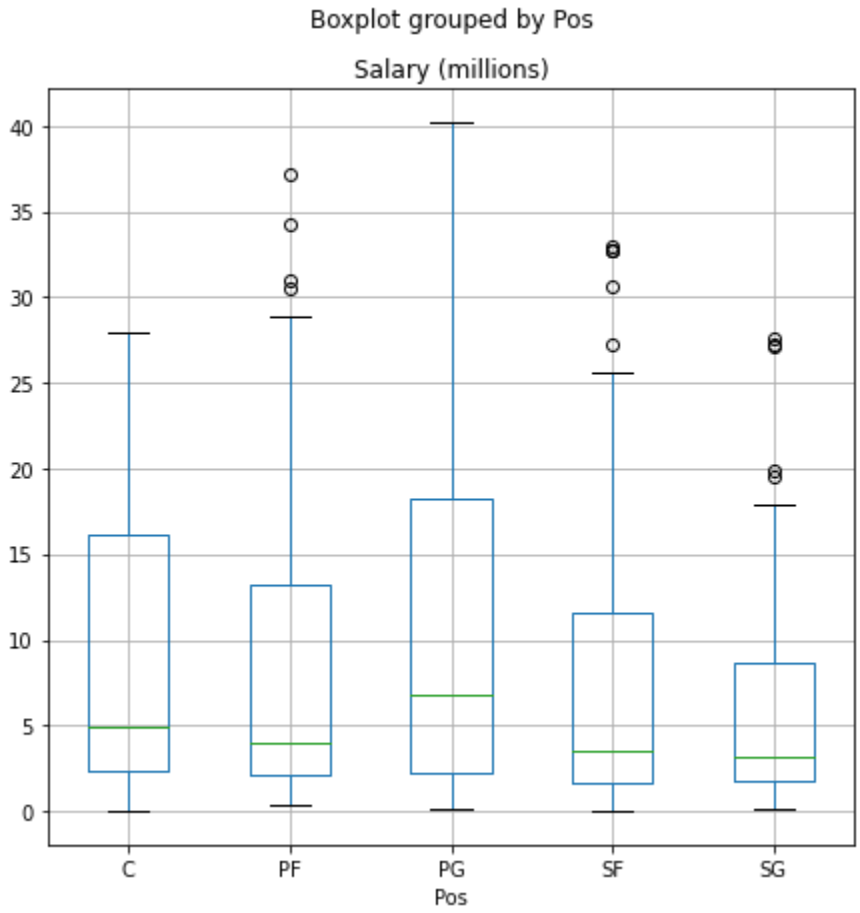

Figure 2: Distribution of contract values at each position

After understanding the positions and distribution of contract values in the dataset, the next thing to investigate was if there was any obvious relationship between a single player statistic and that player's contract value. First, a few of the most commonly reported player statistics (points, assists, and turnovers) were plotted against contract value to see if there was a clear visual correlation (see Figures 3, 4, and 5).

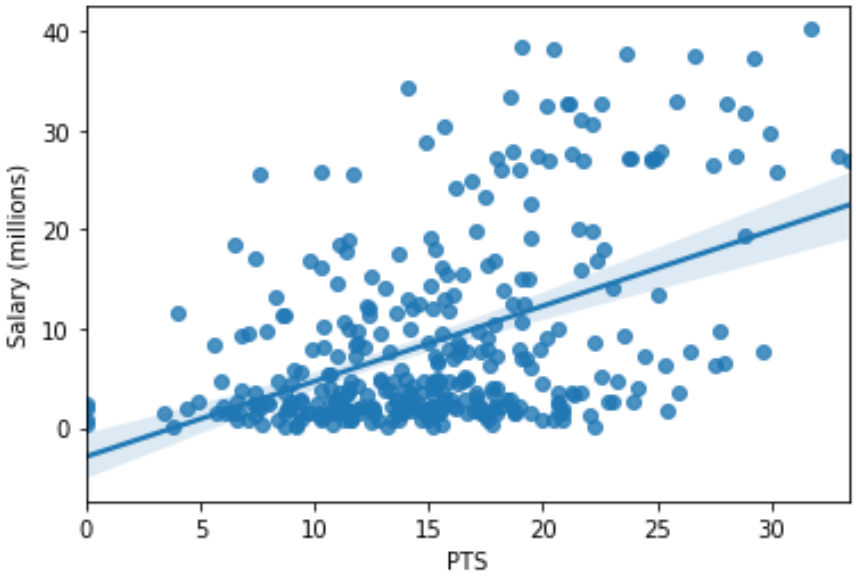

Figure 3: Scatter plot of Points vs Player Salary with regression line.

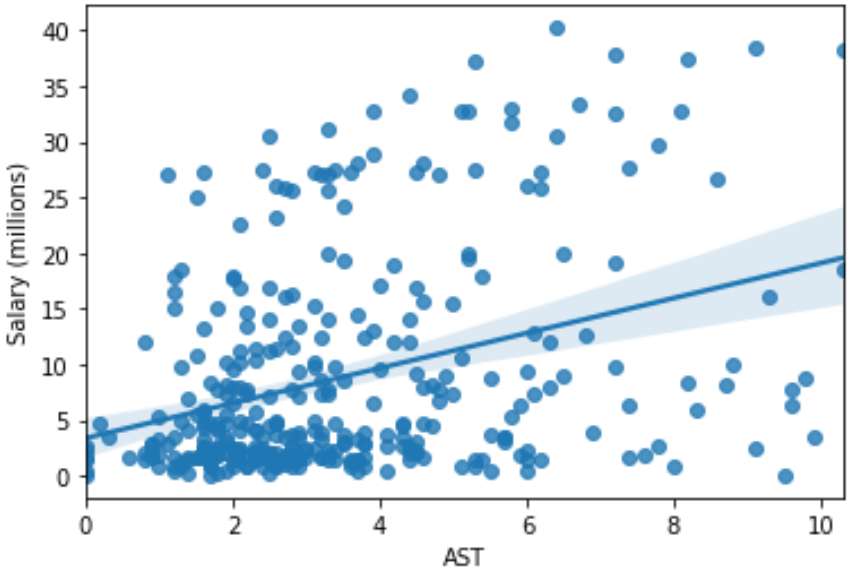

Figure 4: Scatter plot of Assists vs Player Salary with regression line.

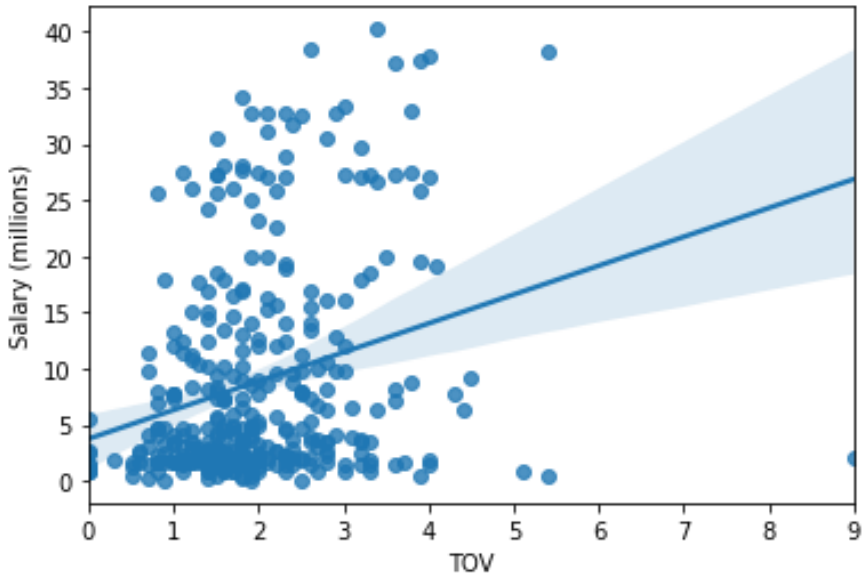

Figure 5: Scatter plot of Turnovers vs Player Salary. Note that an outlier (TOV=30) was removed.

Each of the plots shows a weak positive correlation, but a large cluster of low salary players with a wide range of player statistics is present in each plot. After analyzing the distribution of minutes played and games started, it was identified that many players had limited game time. This is a potential reason for the variability in player statistics for the low salary players who generally do not play as much.

A correlation heat map was generated to show the top 10 variables correlated with a player's contract value (see Figure 6). Interestingly, nothing alone seems to have a strong correlation with player contract value. Note that this does not include the categorical variables like contract type or team.

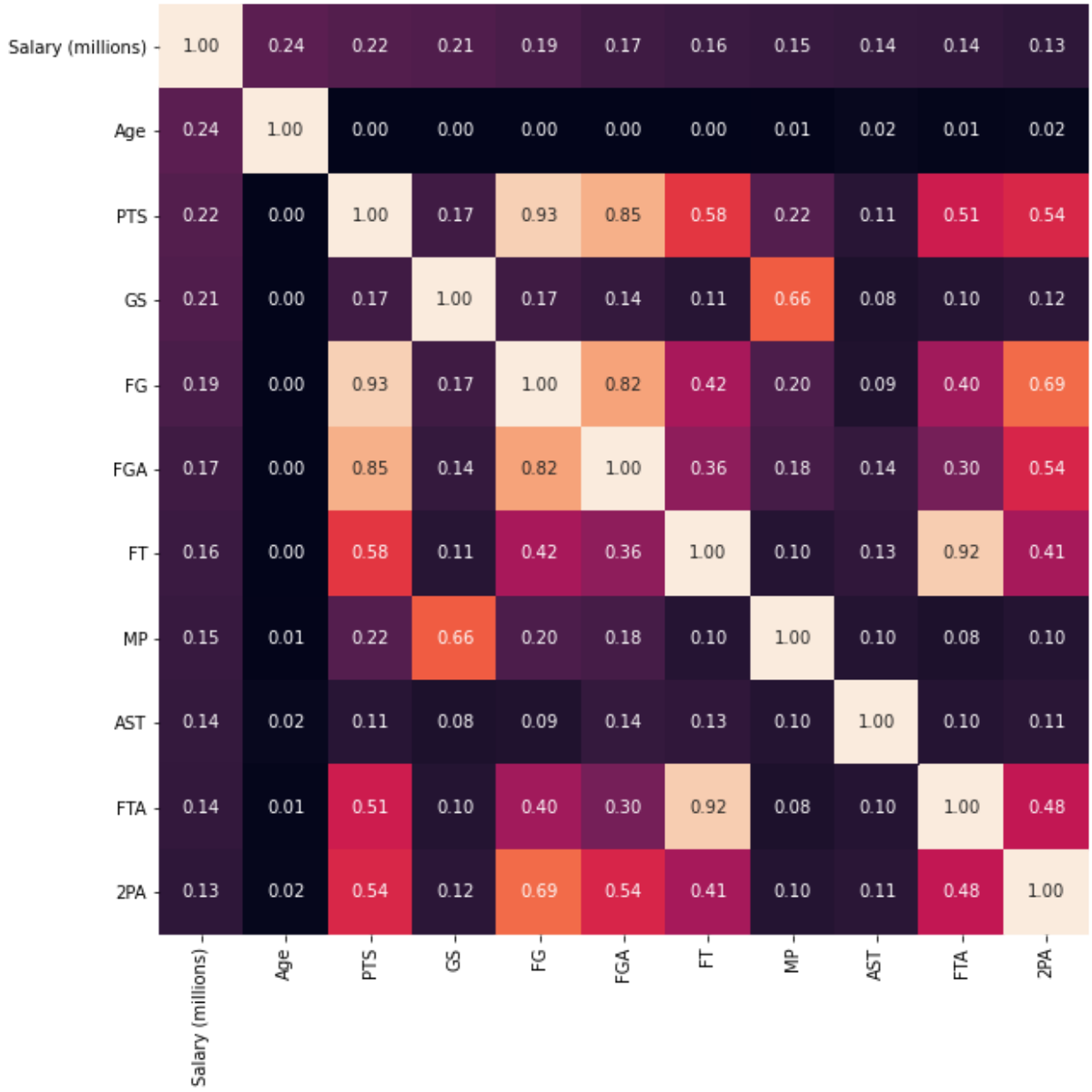

Figure 6: Correlation heat map showing relationships between variables.

Lastly, the distribution of contract values for the different contract types was plotted. The key insight from this distribution was that there are more than 60 1st Round Picks earning less than 5 million dollars. Compared to the other types of contracts, this distribution is the most skewed. There is a limit for the first overall pick in the NBA draft each year, with all the following draftees earning less and less as drafting continues. For these rookie players, it is possible that they are not earning what they

deserve based on their performance because they are still locked in their initial contracts.

Model Selection

A variety of models were tested: Linear Regression, Ridge, Lasso, K-Nearest Neighbor, and Random Forest. To compare the model's performance, the r^2 value, root mean squared error (RMSE), and mean absolute error (MAE) were determined for each model. The r^2 value was calculated for y_pred vs y_test, where $r^2=1$ means all predicted values were equal to the actual values.

Linear Regression

The basic linear regression model did not perform very well. An obvious issue with the predictions was that there were negative values, which would mean a player was losing money (see Figure 7).

$r^2 = 0.23$
RMSE = $7.70 million
MAE = $6.41 million

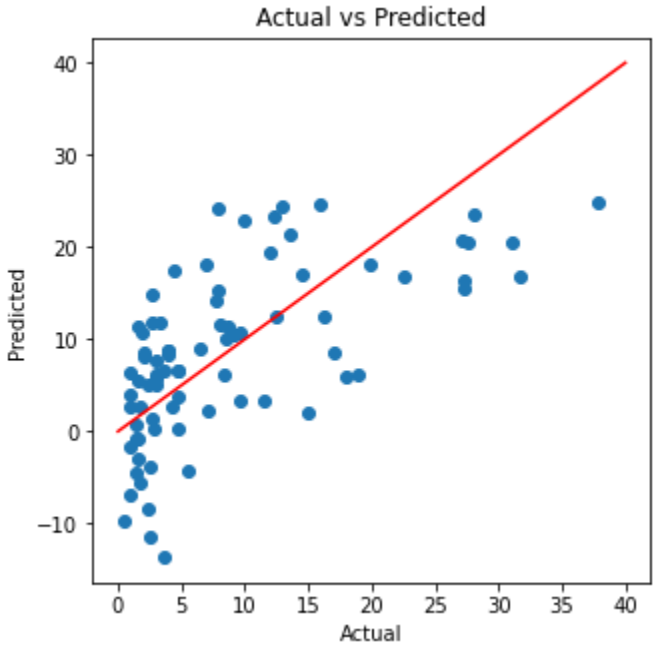

Figure 7: Comparison of predicted contract values with actual contract values for the Linear Regression model. Note the red line has slope=1, where the predicted values are equal to the actual values.

Ridge

Before using the model, the alpha parameter was optimized for the training data. Many alpha values were tested and the value that resulted in the highest r^2 and lowest RMSE value was selected (see Figure 8).

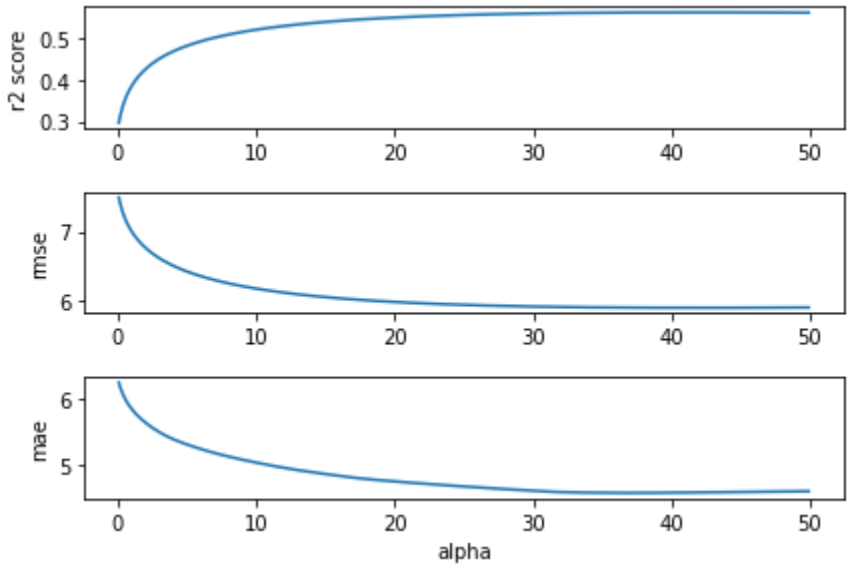

Figure 8: Plots of r^2, RMSE, and MAE for different alpha values.

Overall the Ridge model performed much better than the basic Linear Regression model in terms of the three model scoring criteria. However, it still predicted negative values (see Figure 9).

$r^2 = 0.52$
RMSE = $6.06 million
MAE = $4.72 million

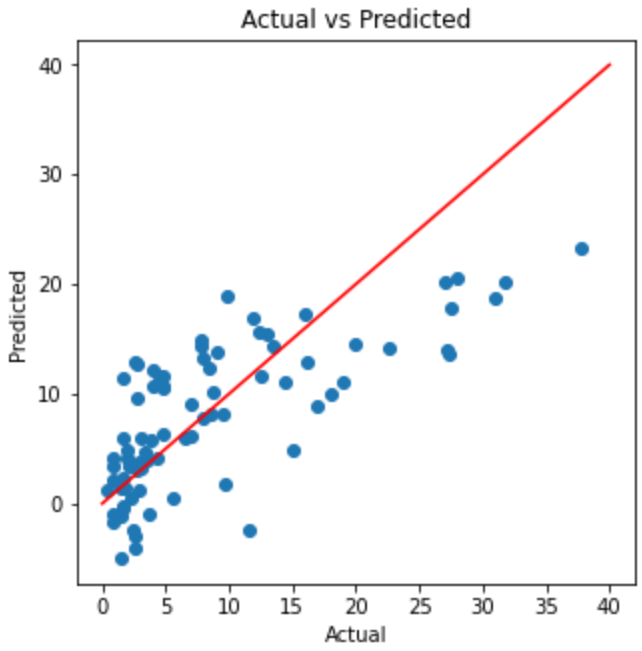

Figure 9: Comparison of predicted contract values with actual contract values for the Ridge model. Note the red line has slope=1, where the predicted values are equal to the actual values.

Lasso

As with the Ridge model, the alpha parameter needed to be optimized before using the Lasso model (see Figure 10).

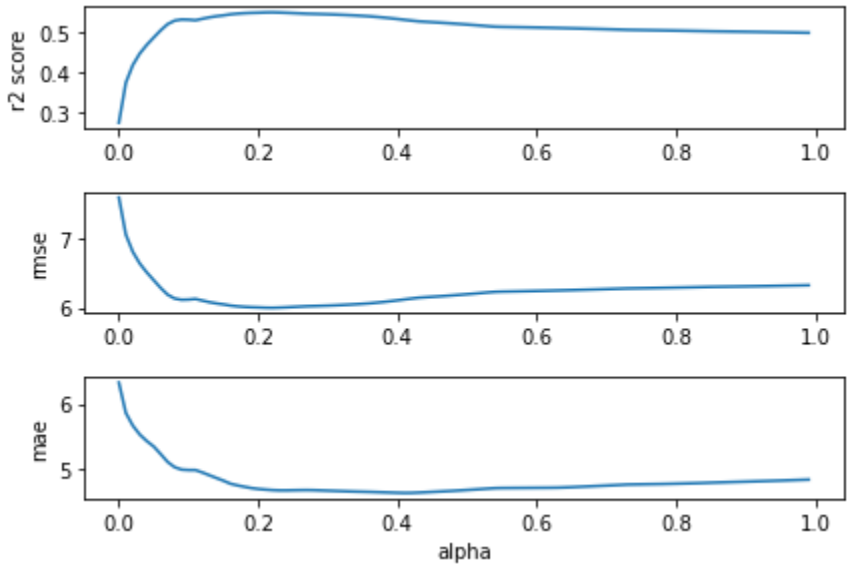

Figure 10: Plots of r^2, RMSE, and MAE for different alpha values.

The Lasso model outperformed the basic Linear Regression model but performed slightly worse than the Ridge model. Just like the other two, this model predicted negative values (see Figure 11).

$r^2 = 0.48$
RMSE = $6.31 million
MAE = $5.12 million

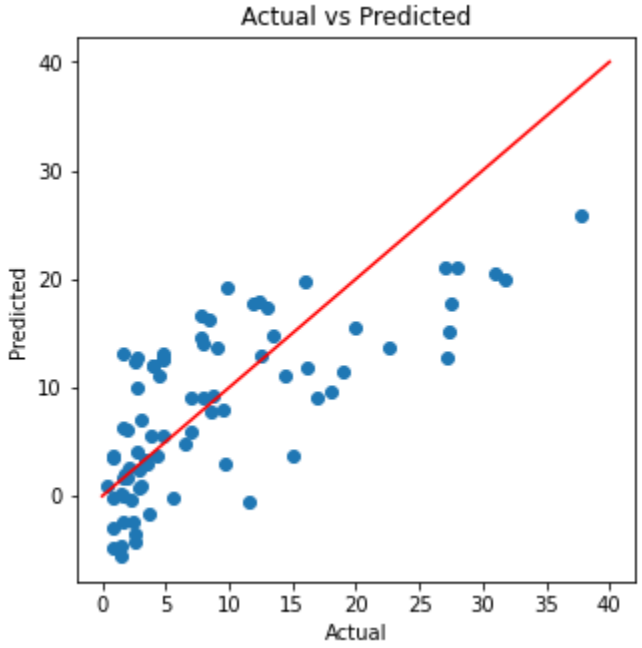

Figure 11: Comparison of predicted contract values with actual contract values for the Lasso model. Note the red line has slope=1, where the predicted values are equal to the actual values.

K-Nearest Neighbors (KNN)

Different k values can change model performance significantly. Similar to the previous models, the k value was optimized using the training data before being used to predict the test values (see Figure 12).

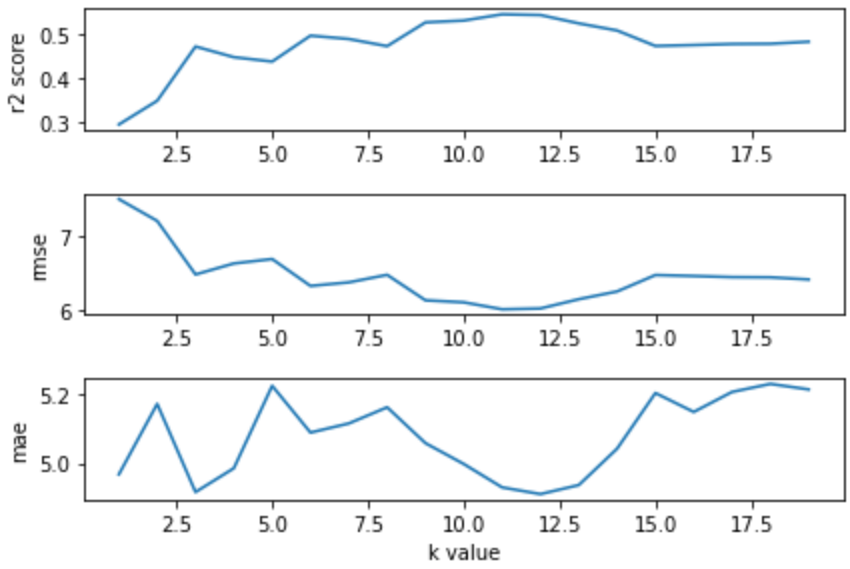

Figure 12: Plots of r^2, RMSE, and MAE for different k values.

In general, KNN did better at predicting low values but did not do well with higher values (see Figure 13). This is likely due to using k=11 neighbors since the highest earning players do not have many close neighbors, so to get to 11 the model has to use lower earning players. The KNN model eliminated the problem of negative predicted values but had lower scoring values when compared to Ridge and Lasso.

r^2 = 0.41
RMSE = $6.71 million
MAE = $4.94 million

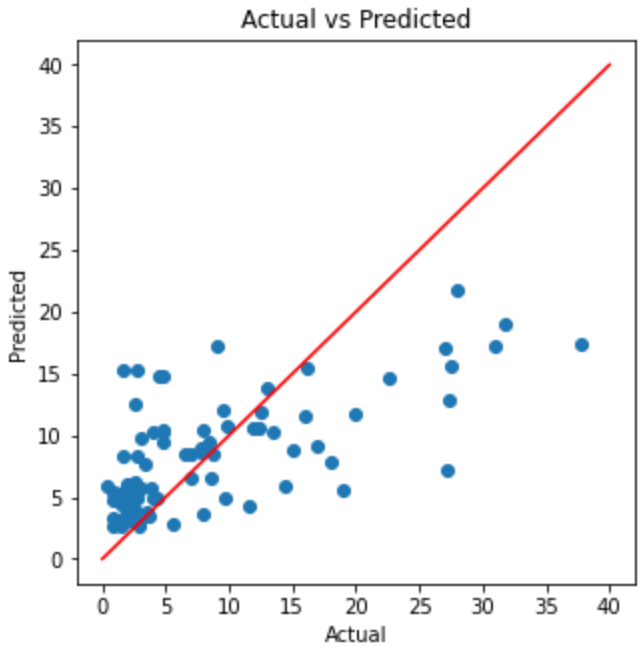

Figure 13: Comparison of predicted contract values with actual contract values for the KNN model. Note the red line has slope=1, where the predicted values are equal to the actual values.

Random Forest

The final model tested was the Random Forest model. Several parameters were optimized using RandomizedSearchCV to fit 3 folds for each of 100 candidates, totalling 300 fits. The best parameters were,

- n_estimators=200
- min_samples_spli=5
- min_samples_leaf=4
- max_features=auto
- max_depth=90
- bootstrap=True

Using the above values, the model performed nearly as well as the Ridge model when comparing the scoring values. Like the KNN model, the Random Forest model did not predict any negative values (see Figure 14). The Random Forest model was also slightly better at predicting high values than the KNN model.

$r^2 = 0.48$
RMSE = $6.34 million
MAE = $4.76 million

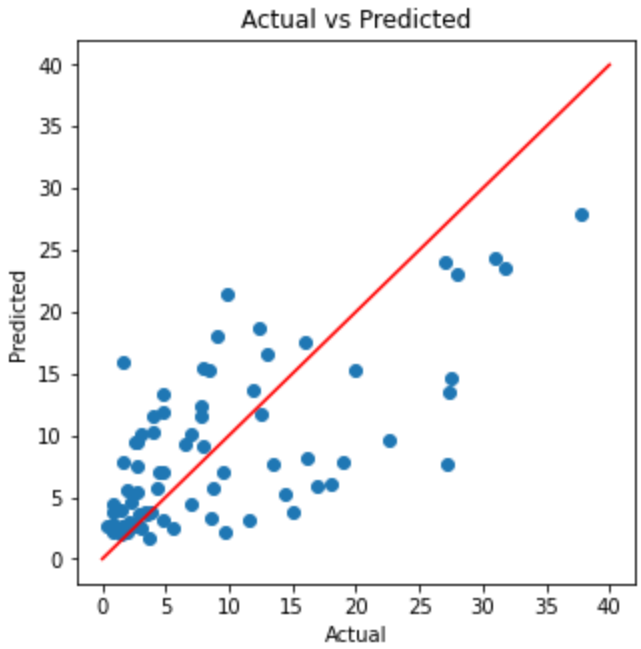

Figure 14: Comparison of predicted contract values with actual contract values for the Random Forest model. Note the red line has slope=1, where the predicted values are equal to the actual values.

Best Model

The best model using only the three scoring metrics would be the Ridge model (see Table 1). However, the Ridge model only slightly outperformed the Random Forest model and it included several negative predictions, which is not acceptable.

Table 1. Comparison of scoring metrics for each model.

	Linear Regression	Ridge	Lasso	KNN	Random Forest
r^2	0.23	0.52	0.48	0.41	0.48
RMSE	7.70	6.06	6.31	6.71	6.34
MAE	6.41	4.72	5.12	4.94	4.76

With this information considered, the Random Forest model would be the best model to continue using. It had the best performance without any gross errors (negative predictions) and was able to reasonably predict both high and low contract values. The feature importance for this model shows that there were two standout features: games

started (GS) and Age (see Figure 15). This makes sense based on the correlation heat map that was generated in the EDA portion of this project.

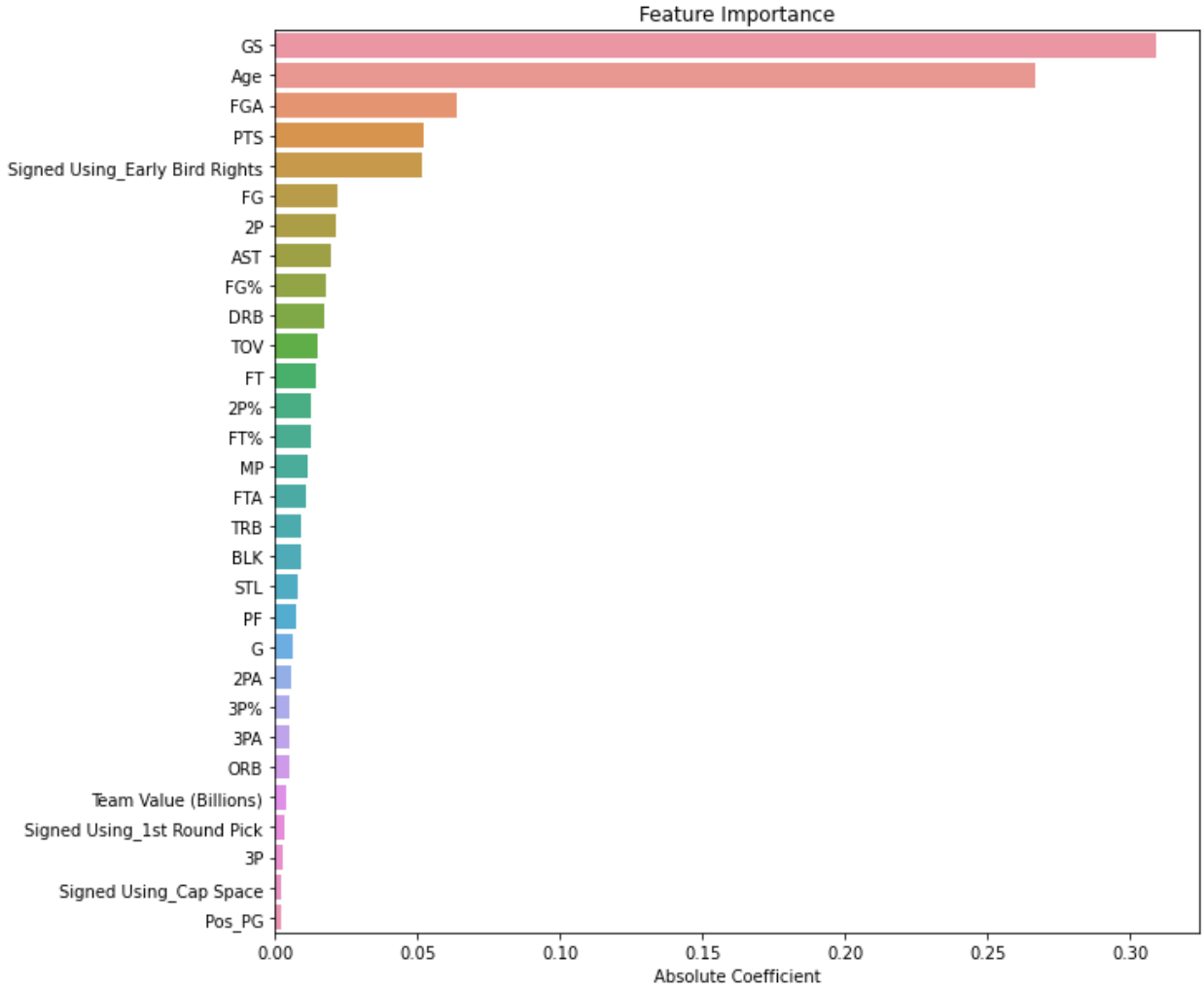

Figure 15: Feature importance for the Random Forest model.

Recommendations

NBA teams and managers should use this model to evaluate their own players before contract re-negotiation, to determine appropriate trades, or to maximize the use of their funds by aiming to sign players with similar performance but lower predicted value based on non-performance traits, such as age or contract type.

NBA players and their agents should use this model to evaluate their own performance and value before entering into negotiations with a team. In addition, if a

player wants to earn more, he can use this model to determine how much better he needs to get in certain areas in order to make a set amount of money.

Future Improvements

If this project were to be continued it would require data from multiple seasons, with monetary values adjusted as needed. Particularly for players with high contract values there is not a lot of data. The KNN model performed well for what it had, but with more neighbors at high contract values it would be expected to perform much better. In fact, all of the models could benefit from more data. The player data was very dense at low contract values and rather sparse at higher values so that made it difficult to predict the high earners. It would also be interesting to see the feature importance for different years since the style of play changes and certain features become more or less valuable to reflect that.

Another possible change would be to use raw player statistics instead of per 36 statistics. The intention of using per 36 statistics was to remove any bias like coach preferences, injuries, or other factors that could result in limited game time for a good player. While this may have worked to an extent, it also introduced the issue of players with very little game time becoming outliers in certain statistics. For example, a low earning player might go into a blowout game with 4 minutes left when all the starters have stopped playing. If this player scores 3 points in those 4 minutes, his per 36 points will be 27. This type of situation can throw off the correlation between important statistics and contract values.