ISYE 6740 - Computational Data Analytics

NBA Salary Analysis and Prediction

Shivani Garg (903209912), Ty Martz (903662133), Teng Zhao (903614965)

Problem Statement

Basketball is one of the most popular sports in North America and is experiencing continuous growth across other regions such as Europe and Asia. The popularity of the sport is partially caused by the star power of the players who compete and how they are able to build their own brands outside of games.This sport is one in which individual players can have powerful monetary and popularity effects for a team and therefore they can earn seemingly excessive salaries.

These player salaries range from a couple hundred thousand dollars to more than 50 million dollars per season, but the players themselves may perform at very different levels statistically. By stripping away the individual player brand, team name, and star power, can we predict what a player should be paid with reasonable accuracy? With information on in-game player statistics, we plan to analyze league data over 10 seasons and understand player salaries based on their performance on the court alone and the influence of star power on player pay.

Data Source

The dataset we are using contains aggregate statistics for individual players during ten years of NBA seasons (2008-2017), including basic player description such as salary, age, position, team, and performance attributes such as points, assists, and rebounds. The original dataset spanned a larger set of seasons, but we decided to narrow in on a ten year span to avoid certain factors, such as inflation, from becoming an overwhelmingly significant contributor to predicting salary. The final version of the data used for our analysis has 4503 rows, each representing a player's statistics for a season, and 55 feature columns that we will use to predict salary. The data was obtained from Kaggle, and originally scraped from Basketball-reference.

Methodology

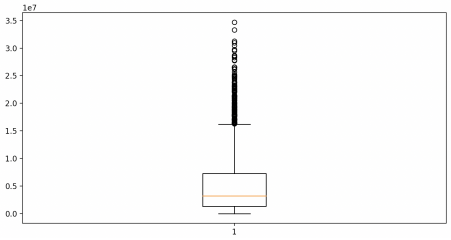
With our inputs being player statistics, we analyzed and preprocessed the data to improve and standardize the inputs in our models. Given that our goal is to predict salaries, there are a number of models we tested and evaluated which include regression, tree-based, and boosting models. Regression models estimate the relationship between the chosen features and the

response variable, in this case player salary. Tree-based models have decision points for each feature to decide where each datapoint falls. Since tree-based models have these thresholds for each feature, these models can trace salary ranges to specific features at certain tipping points offering more insight into which values for each factor specifically influence salary. A risk with these models is overfitting, so the parameters must be well tuned. Ensembles of smaller models are what make boosted models powerful. Using the collective power of many, boosted models sequentially train and evaluate small models with each subsequent one attempting to correct for the flaws of the prior. We will evaluate the results for each of the models to determine which was able to most effectively predict salary.

Data Preprocessing

In preparation for using the NBA player data to train our models, we had to groom the data to better suit the requirements of the models. The first step in preparing our data was to remove the player identifying columns such as player ID, player name, position, and team. This removes information from players since they can have similar annual salaries over the span of a few seasons. We decided to handle the NA/null values by filling them with the mean of that column, although there were very few missing values in the first place.

Knowing that there is the potential of outliers in the salary data, we decided to plot it on a boxplot to identify those outliers. Based on the boxplot visual, we determined that the majority of the salaries lie between $300,000 and $2,000,000 so we further filtered our data to only include those salaries. Before filtering the data based on these salary thresholds, the models had Mean Absolute Errors (MAEs) that were almost the same as, or greater than, the actual salary value. After this filter, the MAEs were reduced to about 200,000. Based on these results, we can infer that the salaries within the filtered range are based more on performance, while the salaries above $2,000,000 are beyond merely analyzing performance; they are related to other factors like marketing value and leadership.



Distribution of Player Salaries

It was also important to standardize the many features to unit variance. We initially started out with 50 columns and thought we should try applying PCA decomposition to identify the most influential features of the dataset. In doing so, we found that only 17 of those 50 columns had an explained variance ratio above 0.01. We filtered our dataset down to these 17 columns. Down the line we noticed the MAEs of the various models increased by including this PCA decomposition. We thought it would be best to remove the PCA decomposition in our further analysis of the data.

We did find that performing a log transformation on our response variable, salary, did improve the MAEs of the models. This helped create a response distribution much closer to normal.

Model Evaluation

With the above methodology in mind, we began to create a number of models. You will find more detail below about regression, tree, and boosting techniques for feature selection and prediction. We focused on the Mean Absolute Error (MAE) metric for scoring while testing parameters and performance between the models. As a minimum means of confirming our model performance, we built a mean salary prediction model, meaning that no matter the feature inputs, the output of the models will always be the same, in this case, the mean of player salaries. As long as our models have a better MAE than this model, they are performing at a bare minimum expected level.

Linear Regression

Our simple ordinary least squares linear regression model includes a large number of input features and since the output is not exactly normal or linear, the model struggles to be the most accurate. When analyzing the fit and output of the model, the coefficients determine the strength of each feature on the output. The typical linear regression model follows this formula: $Y_{hat} = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p$ where b_i is the i^{th} coefficient and X_i is the i^{th} feature. b_0 = intercept and p is the number of features.

The top positive coefficient was Box Plus Minus which is a metric created to score overall player performance on the court. This metric was the largest for both the full and filtered dataset. One difference between the two datasets was that playing time per game contributed more to positive salary change on the full dataset. This shows that the star players tend to get more playing time, which in turn allows them to earn greater statistics and earn more money. The greatest negative coefficients were mostly defense oriented which shows that defensive ability does not contribute to a greater salary and players may focus less on those skills.

Random Forest

Random Forests use an ensemble of decision trees to make a prediction. Each decision tree is fit on a random training set of the data. The randomness allows for more variety in output so that when the predictions of each tree are averaged together, they end up with a good estimate of the answer.

We tuned our random forest by pruning the depth of the trees and choosing the number of trees to be included in the forest. This created one of the better performing models. Overall, Random Forest models gain accuracy with the power of many simpler models and including some randomness in the input.

XGBoost

XGBoost is similar to a Random Forest in that it involves an ensemble of decision trees. The difference is that it uses gradient boosting in an attempt to improve the output. Gradient boosting involves the weak learners to fit the data in a subsequent fashion and improve on the poor parts of the prior one's performance. The target for each model is updated in order to optimize and cover the cases missed in prior models. This is robust and uses the power of many to cover multiple different predictions. This model was one of our best, no matter the filters on the data or whether it was tuned or not.

Ridge Regression

The NBA player data includes more than 50 features to evaluate each player. Some of those features are highly correlated such as "3-point field goal percentage" and "3-point field goals". To mitigate the noise from multicollinearity, which increases model variances, we believe a Ridge Regression model will be a good fit. By performing L2 regularization, the Ridge Regression model shrinks the parameters. Therefore, it prevents multicollinearity and reduces model complexity.

We have also tested the Elastic Net model and LASSO model. We are curious if the LASSO model can help us select some major features which explain the most of variance; or the Elastic Net model can have better performance by combining both L1 and L2 regularization. However, those two models have large MAE. For this data set, we found that the Ridge Regression model has one of the best MAEs among all models we run.

Support Vector Regression

We have tested many linear regression models, although the relationship between salary and player performance stats may not be perfectly linear. Support Vector Machines (SVM), as a machine learning method, are built to analyze nonlinear data by applying the kernel trick to fit the maximum-margin hyperplanes in a transformed feature space. Support Vector Regression (SVR) uses the same principle as SVM, but for regression problems. When the SVR algorithm moves along the data points, it creates decision boundaries to examine each data point. And the best fit line is the hyperplane that has a maximum number of points covered.

We have tested several kernels including linear kernel, Gaussian radial basis function (rbf) kernel, and polynomial kernel. Overall, for this data set, we see SVR with the rbf kernel has one of the best MAEs among all other SVR models we run. It performs particularly well on the full dataset in comparison to the rest of the models we mentioned. We are not surprised to see the adaptivity of machine learning models can deliver very good prediction performance for complex nonlinear data.

Final Results

Model	Base MAE Score	Tuned MAE Score	Change in MAE
Support Vector Regression	2300525.67	2295700.45	-4825.21
XGBoost	2371932.13	2313584.12	-58348.01
Random Forest	2371391.04	2357459.39	-13931.65
Ridge Regression	2578427.90	2567439.61	-10988.30
Linear Regression	2580448.27	2580448.27	0.00
Mean	5383383.87	5383383.87	0.00

Full Data

Our results above show model performance on the full dataset. The best model using the mean absolute error metric (MAE) is the SVR model. Both with and without tuning, this model predicts the best of all. Close behind were the XGBoost and the Random Forest which performed quite similarly overall. The regression models perform slightly worse, but all models score much better than a mean prediction model. As expected, the nonlinear models perform the best.

Model	Base MAE Score	Tuned MAE Score	Change in MAE
XGBoost	271089.72	265882.79	-5206.93
Random Forest	270423.41	269242.55	-1180.86
Support Vector Regression	279680.21	278937.24	-742.96
Ridge Regression	302926.02	302287.23	-638.79
Linear Regression	304258.29	304258.29	0.00
Mean	1214934.66	1214934.66	0.00

Filtered Data

To see if we could optimize our models performance, we attempted to remove outliers in the dataset. This filtering left us with predicting a more normally distributed salary range of $300k - $2M. Using this data, our models were able to achieve much improved MAE scores, although we must take into account that the range of potential responses is much smaller in proportion to the full dataset. The improved scores show that much of the larger salaries may be attributed to things other than performance statistics. This could include marketing value, which brings in more revenue to the team in ticket and merchandise sales.

Conclusion

Given more time, we may want to explore more features to help in predicting salaries. This could include pulling data other than game statistics as model inputs. Given what we learned, some of these features could be jersey sales, number of social media followers, and amount of endorsements. Taking into account the social power of the players could potentially help improve the models. Eventually, with a good model and robust data, this could be used to create a player scoring system to see what they are truly worth. This model would allow coaches and team owners to understand the worth of the players with a new metric and make decisions accordingly. Also, with this information, they can manage their player salaries, trades, and retention to optimize for team revenue or player performance.