NBA Player Positions

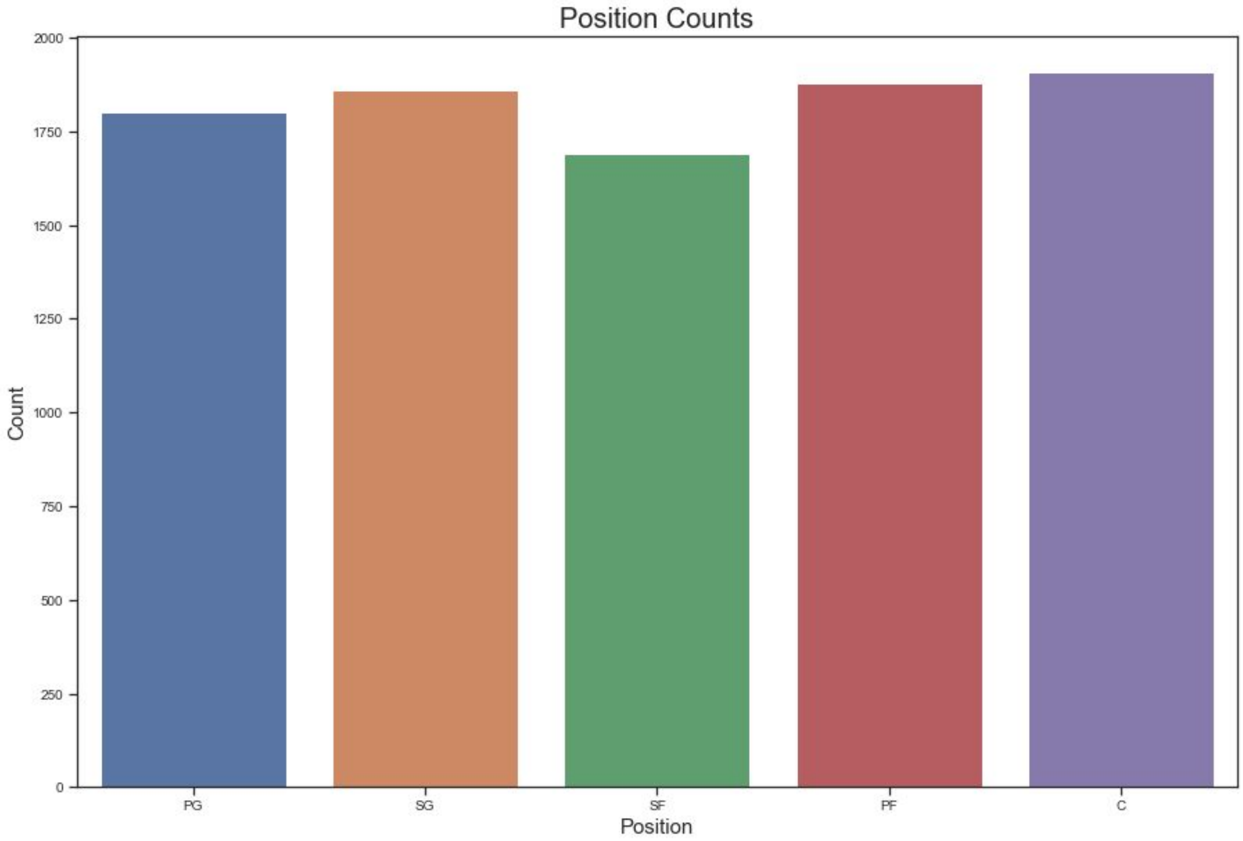
Jason Katz, Brown University,

GitHub: https://github.com/jasonk33/nba-player-positions

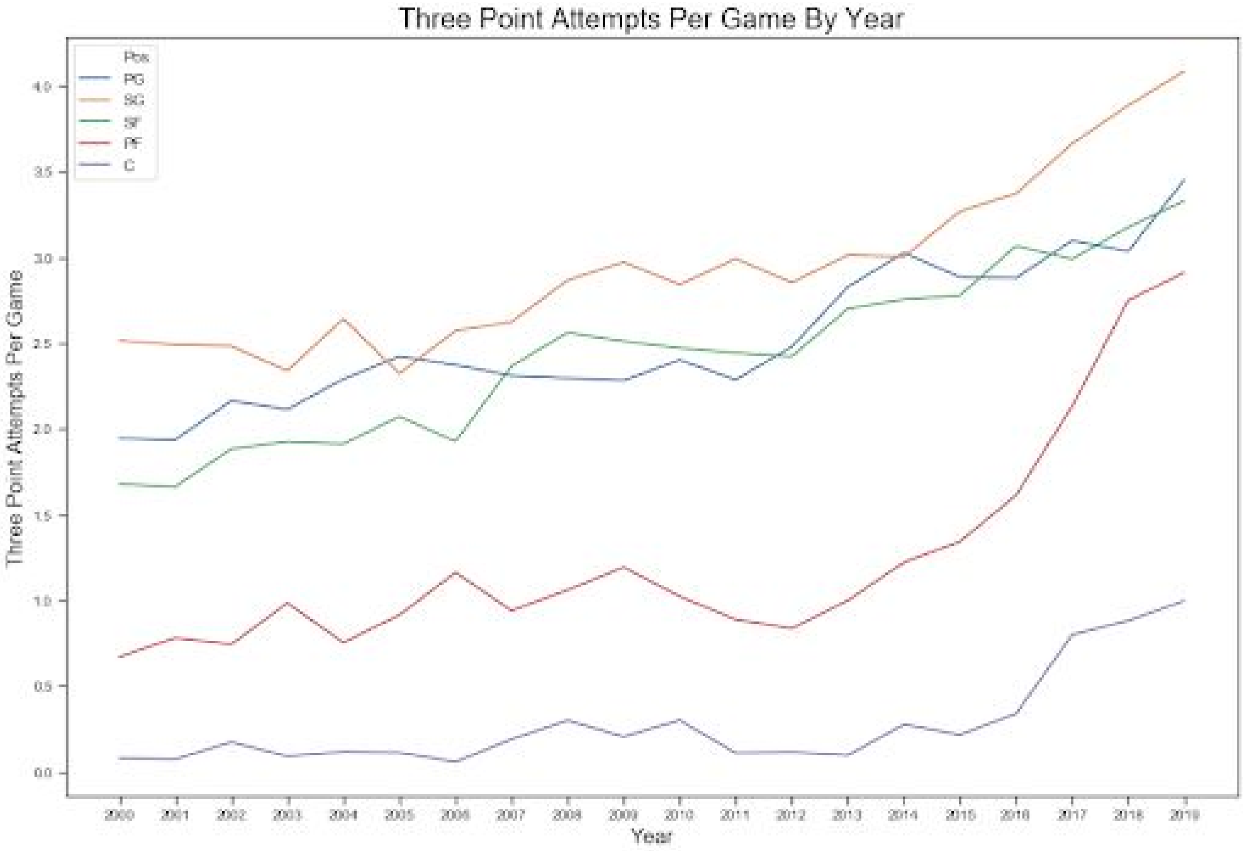
Introduction

For this project, I am trying to gain greater understanding into the different NBA positions. There are 5 positions in the NBA (point guard, shooting guard, small forward, power forward, and center). Each position has a unique skill set and it would be interesting to see which in game statistics contribute most to a player's position designation. This will be a classification task, where I predict a player's position based on their seasonal statistics. This is interesting because many NBA players are thought to be able to play different positions, and understanding what stats go into a player's position designation can give greater insight into the structure of NBA positions and how they have evolved over the years. The dataset for this project was scraped from basketball reference and contains around 10,000 total players from the 2000 through 2019.

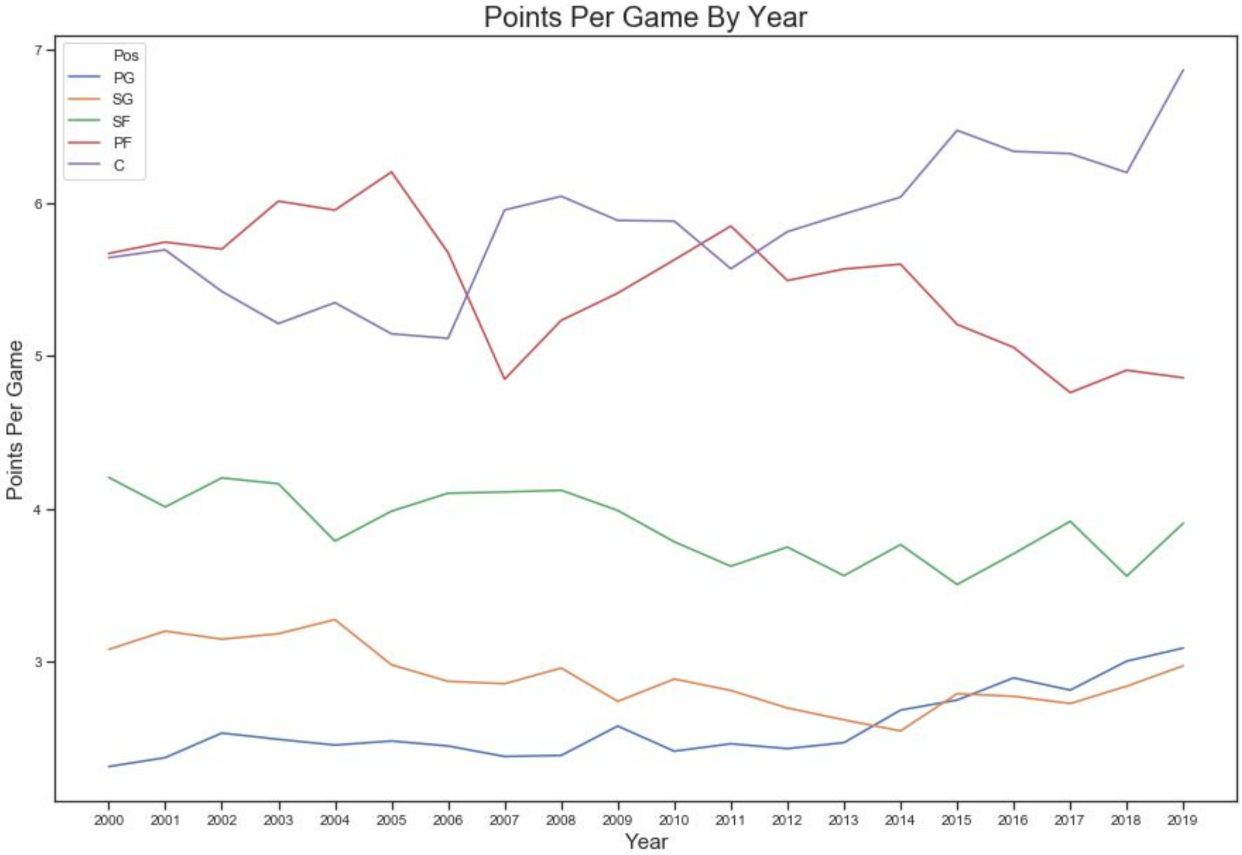
EDA



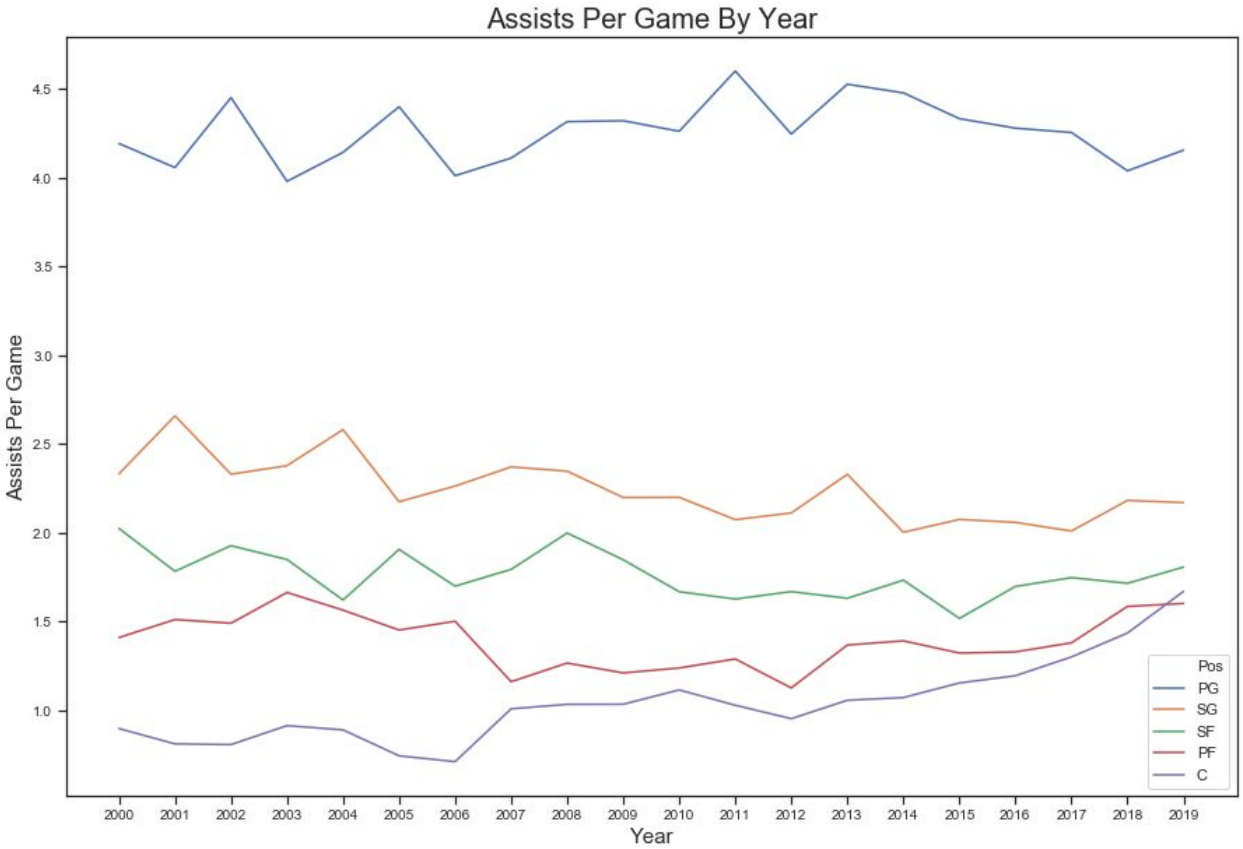
This figure shows the breakdown of number of players for each position in the entire dataset



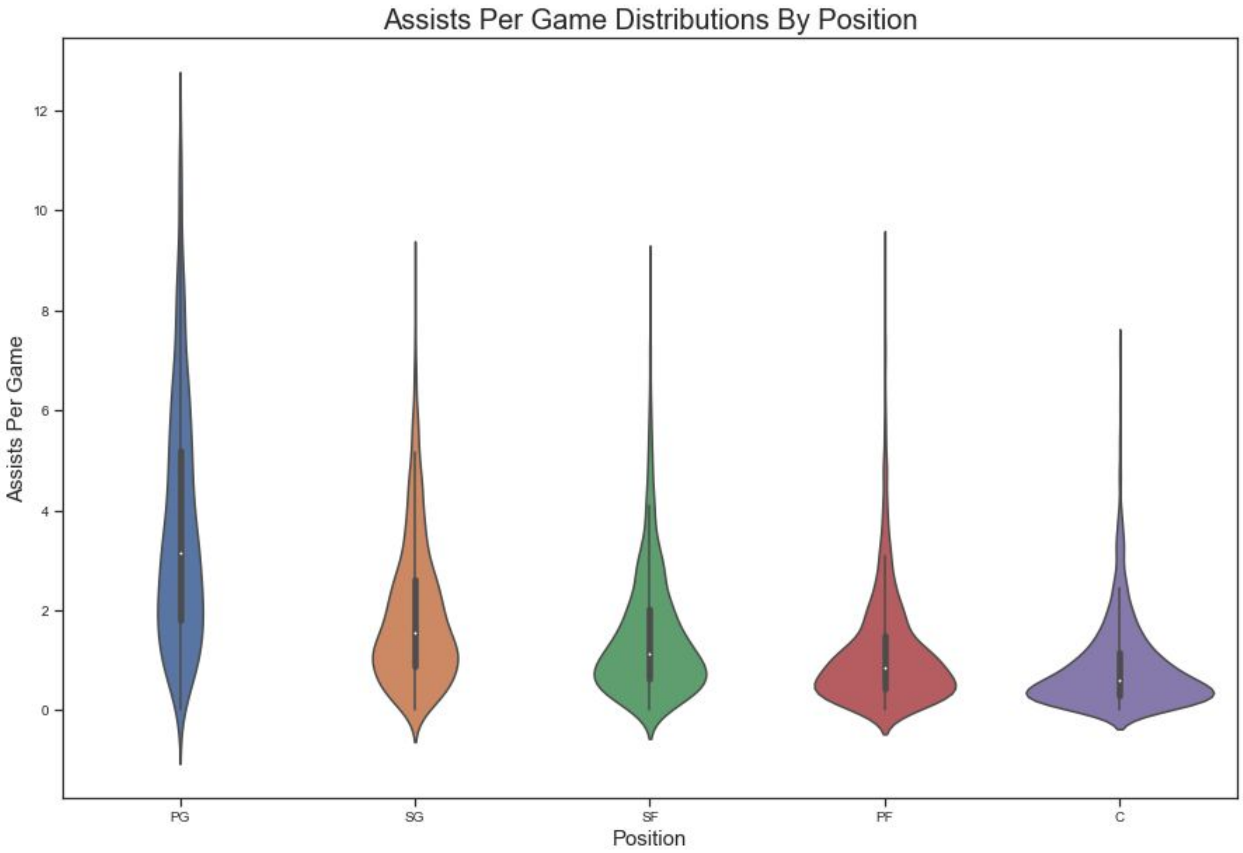
This figure shows how the number of three pointers taken per game has been rising over the years, especially for PF



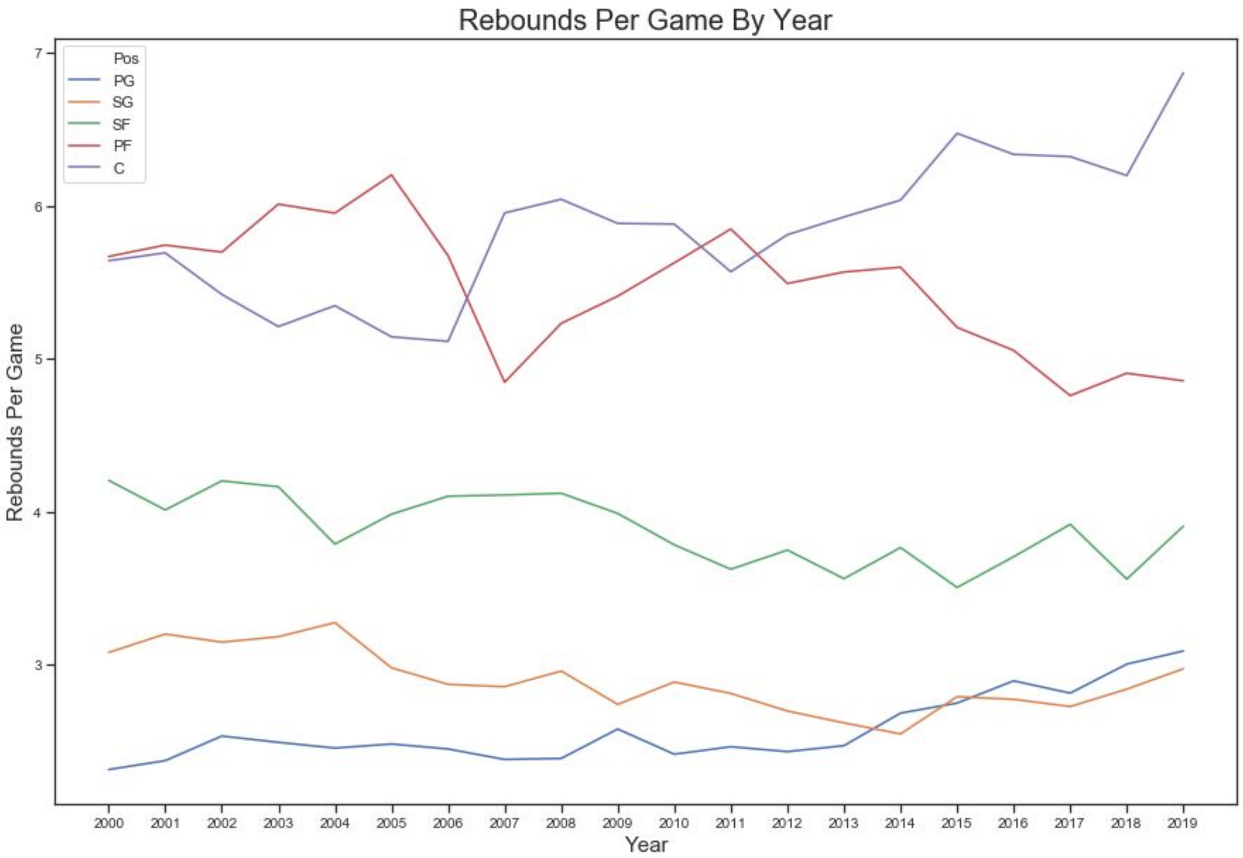
This figure shows how points per game has changed over the years for each position



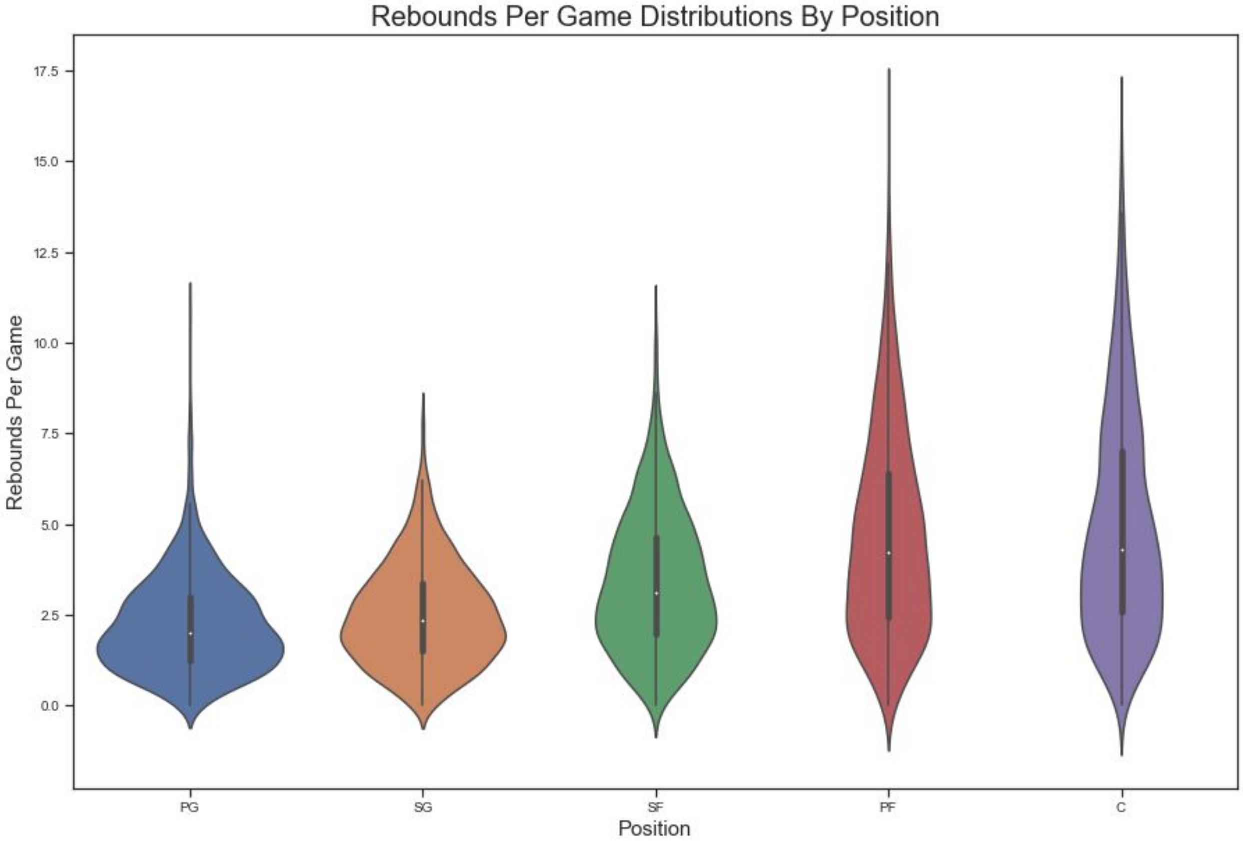
This figure shows assists per game by position over the years, note how point guards have the most, but centers have started to rise in recent years



This figure shows the distribution of assists per game by position, point guards have a very narrow distribution compared to the other positions



This figure shows rebounds per game by position over the years, note how centers have overtaken power forwards



This figure shows the distributions of rebounds per game by position, note how the power forward and center distributions are very narrow compared to the other three positions

Methods

For preprocessing, most of the features were continuous, so standard scaling was applied. One hot encoding was applied to the single categorical feature. The label had multiple values, so a label encoder was used. After preprocessing, there were 22 features and 10,000 data points. For the labels, almost every player has one of the 5 main position designations (PG, SG, SF, PF, C), but a select few players are listed under multiple positions. Since the number of players listed under multiple positions is small, those were removed from the dataset. Players who did not record any minutes were also removed. Players who were traded during the season, were listed more than once, with statistics for each team they played for, as well as their total stats combined for the year. For players that were traded, the rows representing their total stats combined for both teams were kept, and the other rows for them were removed. There were some missing values for field goal percentage statistics when a player had not attempted any shots. Since field goal percentages can be calculated directly from field goal makes and attempts, the percentage features were removed. Since all the statistics were totals for the season, they were each divided by the total number of minutes played to put each stat on a per minute basis.

For the machine learning pipeline, I used repeated k fold stratified cross validation. I chose to use stratification to make sure the classes for each of the five positions were balanced. I used five folds and three repeats for the cross validation. I tried five different kinds of models and tuned the hyperparameters using a grid search. For logistic regression I tuned the value C and tried the range 0.1 to 1.0. For random forest, I tuned the maximum depth and the minimum number of samples for a split, I tried the ranges of 5 to 10 and 3 to 11 respectively. For k nearest neighbors, I tuned the number of neighbors, trying the range 25 to 75. For gradient boosting, I tuned the max depth, the number of estimators, and the learning rate, I tried the ranges 5 to 7, 50 to 100 and .05 to .15 respectively. For support vector machines I tuned C and gamma, trying the ranges of 150 to 250 and 0.001 to 0.1 respectively. I used both accuracy and logistic loss to evaluate the models' performance. I used accuracy as the main metric because it is easily understood, but also made sure to watch logistic loss to ensure the predicted probabilities were reasonable outside of pure class prediction. I used the same random seed for splitting and non deterministic models to ensure all results were reproducible.

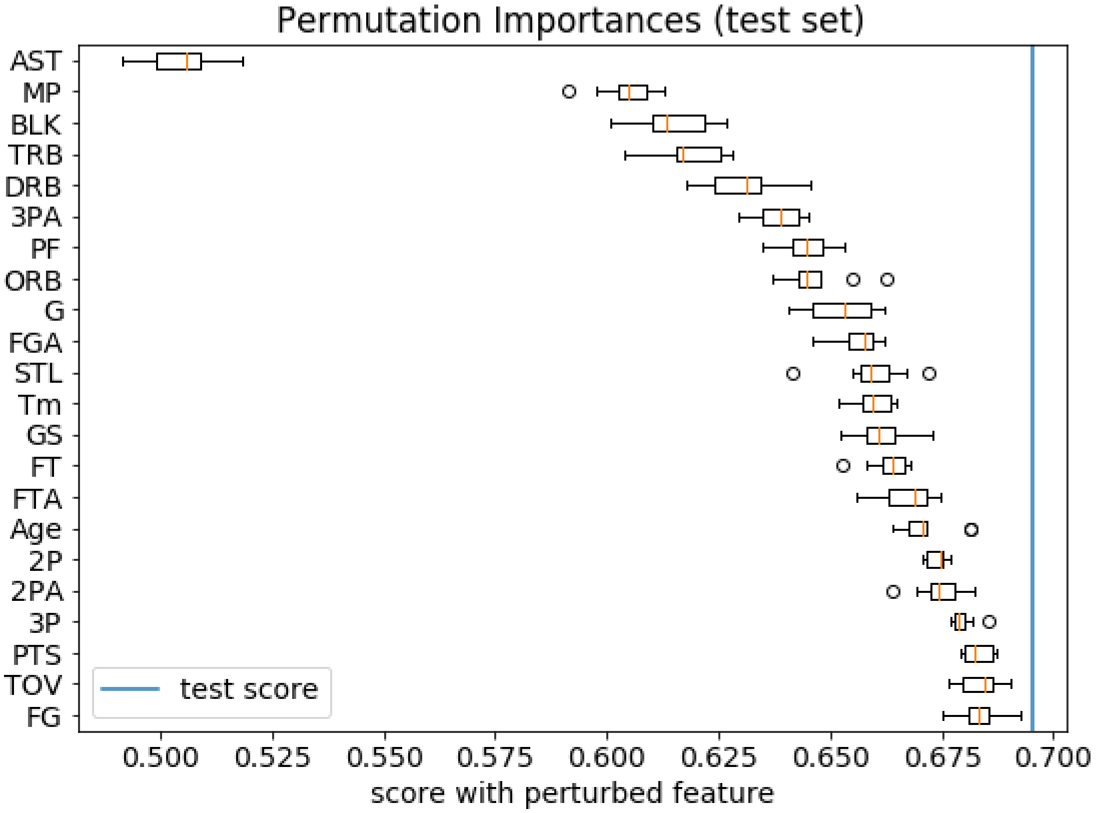
Results

The baseline model for this five class classification task was .209.

	Cross Validation Accuracy/Log Loss	Validation Accuracy
Logistic Regression	.622 / .948	.615
Random Forest	.638 / .886	.636
K Nearest Neighbors	.635 / .921	.631
Gradient Boosting	.662 / .822	.686
Support Vector Machines	**.679 / .762**	**.695**

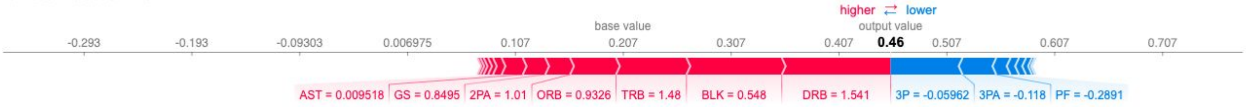
Model results using 5 fold stratified cross validation with 3 repeats

Out of all the models, the one that performed the best was the support vector machine, with an accuracy of .695. Gradient boosting was the second best method.



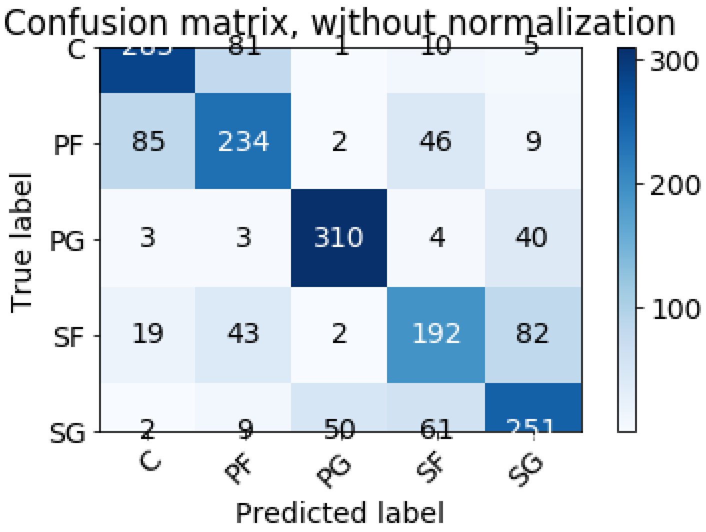
Permutation test for global feature importance from SVC model

Using a permutation test to determine global feature importance, assists were the most important feature by far, which makes sense, as that stat often indicates a point guard (and rarely a big man). Three point attempts, blocks and total rebounds were also important, as all of those indicate it is a big man. Interestingly, minute played is very important, even after using it to standardize the other features. These results fit into a human context, as the most important features all pass the eye test and the ability to make a good predictor reinforce the idea that there is a fundamental difference between the five different positions in what they do on court from a stats perspective.



This figure depicts local feature importance for an example of a center, where the model predicts it correctly

Using shapely, I calculated local feature importance for a single example . The above figure shows which features go into a prediction for a specific example of a center. You can see that high rebounds and blocks push the prediction more to the right.



This figure shows the confusion matrix for the SVC model on the validation set

Looking at the confusion matrix for the best model, which was the SVC on the validation set, we can see that the model does a pretty good job of separating the different positions. Sometimes it confuses centers with power forwards and shooting guards with small forwards, which is not bad since those positions are often similar. It is good, because you can see it rarely mixes up guards with big men.

Outlook

To improve the model, I would experiment with other types of models, such as gradient boosting or neural networks. I would also see if there are any other hyperparameters I could tune, possibly different types of kernels for the support vector machine model. I could also acquire more data in addition to seasonal statistics, such as salary information or player details such as height and weight. I could also experiment with different feature engineering to combine different stats together in different ways.

References

Data was scraped from basketball reference. While doing research for the project, I found three other people who performed similar work, trying to predict player positions:

Predicting NBA Player Positions
NBA Position Predictor
Predicting NBA players positions using Keras