In-Game NBA Stats

Tyler Oh
Jerome Lau
Khaled Abdelaziz

Aug 4, 2023

1. Introduction

Problem

The NBA, National Basketball Association, is one of the largest major professional sports leagues in the world. As sports analytics has improved, many powerful tools have been implemented in order to ensure their team has the highest chance to win the game. In light of this, we decided to perform our own statistical analysis to determine which in game stats contribute most to winning the game. It is already clear and well documented that the team playing on home field has the advantage, thus our aim is to determine which of the statistics has the largest influence on the outcome of the game. We hope to find conclusive evidence for what a team should focus on to improve their win rate.

2. Data

Data-Preprocessing

The provided data we used is from Kaggle's "NBA Database" and the particular dataset we chose was game.csv which includes a wide range of variables for in-game stats. We started by filtering out any unnecessary variables and only selected those that refer to the game statistics. As statistics such as the points, field goals made, free throws made, and assists directly correlate to winning the game, as the team with more points wins, we excluded them. This leaves us with the following stats: field goals attempted (fga), free throws attempted (fta), rebounds (reb), steals (stl), turnovers (tov), and personal fouls (pf).

These variables included the stats for both the home team and the away team so we also created a column where we combined each one. This was done by subtracting the away stats from the home stats to get the difference. We did this so the data would be observed with respect to the home team such that a positive value meant the home team is greater and a negative value indicates that the away team is greater. We then selected to observe only the games which were played in the Regular Season. This was done to ensure that the level of play was consistent as other season types such as Pre-Season might have a lower level of consistent play or Playoffs would only include select teams. Afterwards, any columns that were missing any stats, we decided to drop that game. Lastly, to have a numeric value for our prediction, we converted the home teams win/loss to a 1/0 respectively.

3. Techniques

T-Tests

We first started analyzing the data by performing T-tests on each variable. We wanted to determine if the averages of each team's stats were different to see if they would be important in developing our prediction model. We stated that our null hypothesis to be that the averages are the same. This would indicate that the particular variable would not have statistical importance.

Prediction Models

Then we started building and training different models to predict the game results. We aimed to determine which models yielded the highest score. First, we started by using Gaussian Naive Bayes. The training model was split with X using all our variables and Y using the Win/Loss.

Then we trained Decision Tree Classifier and Random Forest Classifiers with Grid Search CV to determine which parameters would yield the highest score. For the Decision Tree Classifier, we explored various criteria to measure the quality of splits, such as ‘gini’ and ‘entropy’. We also explored various values for parameters such as the maximum tree depth, minimum samples required for node splitting, and minimum required at each leaf node in order to find the best hyperparameter. Similarly, for the Random Forest Classifier, we fine-tuned hyperparameters such as the number of trees, maximum features considered at each split, maximum tree depth, and bootstrap.

Afterwards, we implemented the K-Neighbours Classifier with cross val score to test which parameters yielded the highest score. Then, we applied the K-Neighbours Classifier and utilized cross-validation with multiple folds to evaluate different parameter values. Lastly, we plotted the k-values against cross-validated accuracy in order to find the optimal number of neighbors which had the best performance. Finally, we trained the Gradient Boosting Classifier to determine which parameters would yield the highest score.

Feature Importance

After we determined which models had the highest scores, we then wanted to see which variables were most important in making the predictions. For this we used permutation importance and feature importance.

Linear Regression

Our final step was to test manually if there were any linear relations between a team's win rate and any their stats. To do this, we totalled all their stats and divided it by their total number of games. Then we fitted a line to see if there were any correlations between their win rate and stats.

4. Results

T-Tests

After our initial T-test results, we found that each variable for both winning and losing all had a p-value less than 0.05 so we reject the null hypothesis. This means that all the variables have a different mean, which would indicate that they are all statistically significant in determining the outcome of a game.

Prediction Models

For our prediction models, we used classifiers to see if training the models could predict the outcome of a game accurately. At this stage, we initially checked if any scaling was necessary, however we found that they were not as the classifiers we used were not vulnerable to non-linear relationships or scaling issues. Our analysis is shown further below.

Table 1. Accuracy Score for Classifiers

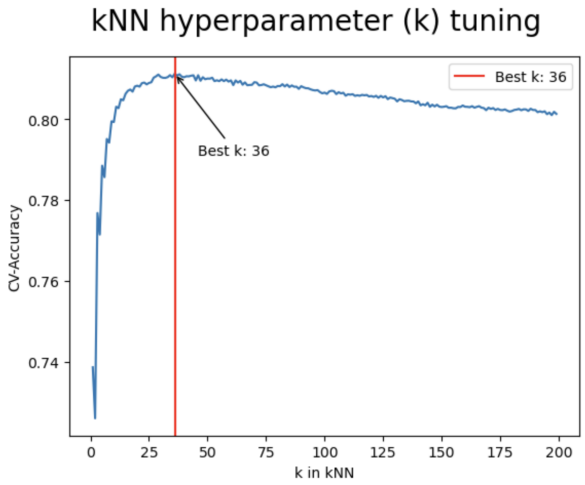
Classifiers	**Training Set**	**Validation Set**
Gaussian NB	0.7295735728191923	0.736906556141673
DecisionTreeClassifier (without hyperparameters)	0.9999057966463606	0.7424642049736248
DecisionTreeClassifier (with hyperparameters)	0.7530616089932801	0.7507535795026375
RandomForestClassifier (without hyperparameters)	0.9999057966463606	0.8084966088922382
RandomForestClassifier (with hyperparameters)	0.7437354769829806	0.7426525998492841
K-NN Classifier	0.8230861018652263	0.8157498116051244
GradientBoostingClassifier	0.8190667587766125	0.8168801808590807

In our classification, we initially employed a basic Gaussian Naive Bayes model. While the GaussianNB model showed no signs of overfitting, its accuracy score was relatively poor. This can be attributed to the GaussianNB assumption of data being normally distributed and independent.

Following this, we proceeded with the DecisionTreeClassifier without setting any hyperparameters. The resulting model shows obvious overfitting, with near-perfect accuracy on the training set but relatively lower accuracy on the validation set. To overcome overfitting, we performed hyperparameter tuning by experimenting with 'max_depth', 'min_samples_split', and 'min_samples_leaf'. In particular, we considered values of 2%, 5%, and 10% of the total number of observations for 'min_samples_split' and 'min_samples_leaf'. This tuning led to an improvement in terms of overfitting, and the model's performance remained slightly better than GaussianNB.

Furthermore, we implemented the RandomForestClassifier without initial parameter settings, which again resulted in a highly overfitted model as DecisionTreeClassifier. To address overfitting, we employed hyperparameter tuning using 'n_estimators' and the same variables in the DecisionTreeClassifier. Despite our effort, the model's performance did not improve significantly and remained relatively worse than even the tuned DecisionTreeClassifier. This suggests that tree-based models might not perform optimally due to the dataset's complexity and size.

Figure 1.



For the next method, we chose the KNeighborsClassifier. Prior to training, we performed grid searching to identify the optimal number of neighbors (k) for the KNN method. We plotted the accuracy for all k-values and determined that the best k-value was 36. Finally, we implemented the GradientBoostingClassifier, one of the robust machine learning techniques. The model showed the best results with notable accuracy scores.

Overall, our prediction models performed fairly well across various methods. Our best validation score was .816, which we consider a notable result. While we recognize the limitations of the dataset, we found the score encouraging our future analysis, considering the utilization of only 6 variables.

Feature Importance

Figure 2.

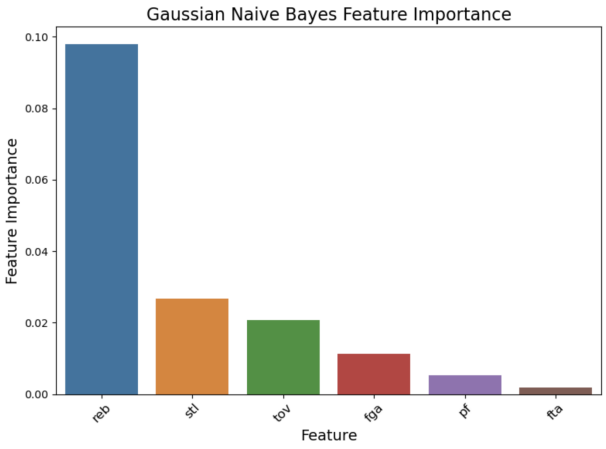


Figure 3.

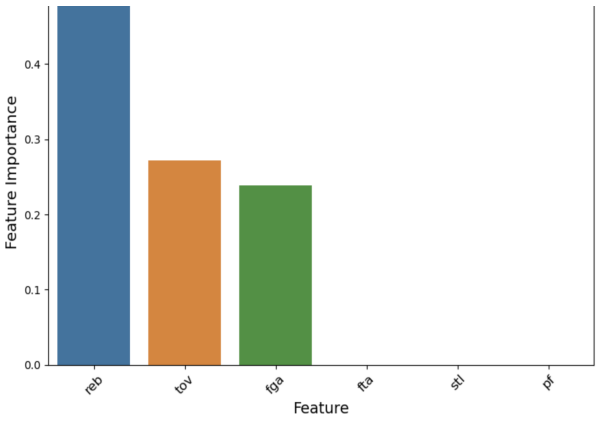


Figure 4.

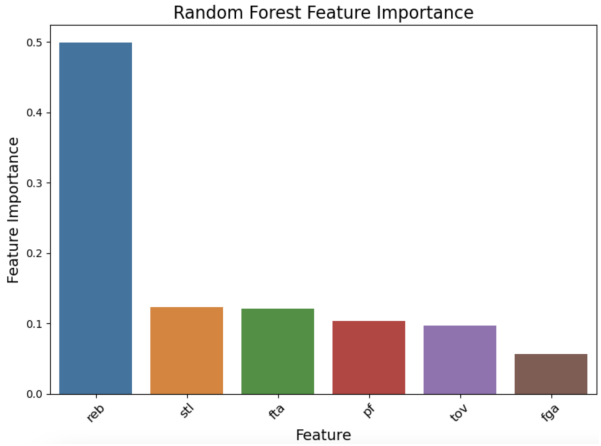
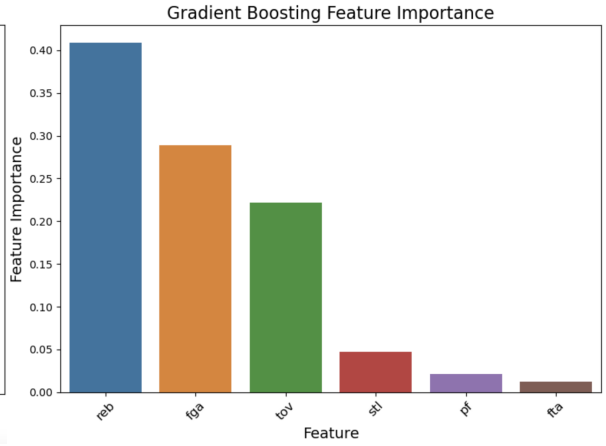


Figure 5.

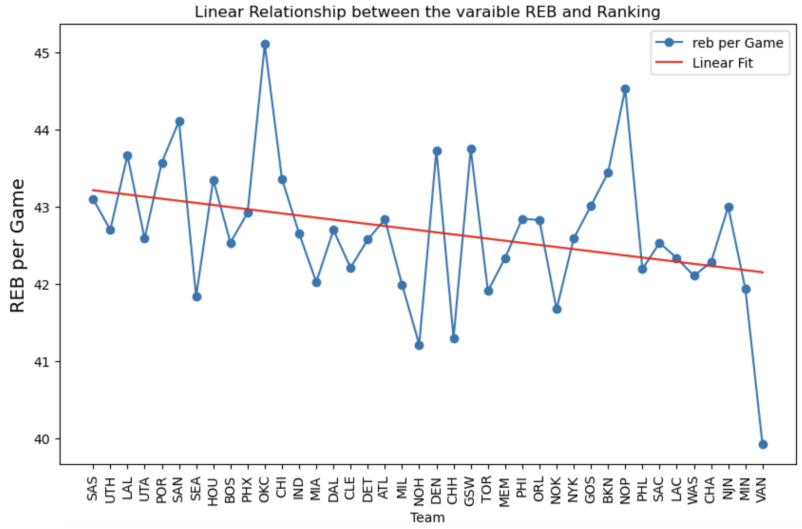


Based on our results from our prediction models, we saw that the models yielded fairly high scores which would allow us to further investigate which feature the model used to make the prediction. For this, we observed Gradient Boosting Classifier, Decision Tree Classifier, Random Forest Classifier, and Gaussian Naive Bayes as they allowed for observing the features that influenced the predictions. As seen in Figures 2, 3, 4, and 5 all the models are congruent in their most important feature used in determining the outcome of a game. This gives us a good indication that rebounds are the highest contributing factor in winning a game. Furthermore, we notice that turnovers also have a significant impact. This is shown in Figure 3 and Figure 5 as turnovers and field goal attempts are the second most prevalent in Gradient Boosting Classifier, which is our best performing model, and Decision Tree Classifier. This gives us conclusive evidence that rebounds is the most important feature in determining the outcome of a game.

Furthermore, we are able to observe that free throw attempts and personal fouls are the lowest impacting variable in the prediction models, with Decision Tree Classifier having no influence at all. While this is not entirely conclusive, we believe that personal fouls and free throw attempts are not meaningful in determining the outcome of a game.

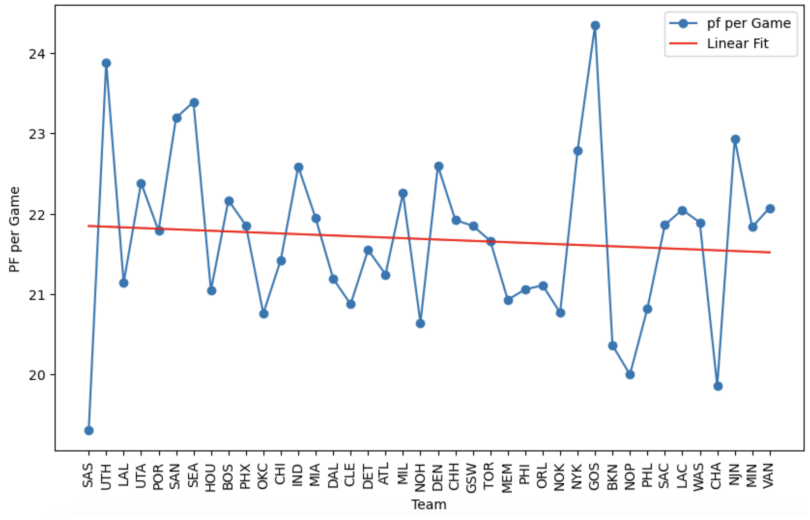
Linear Regression

Figure 6.



As our last step, we wanted to correlate the results from the feature importance of the prediction models and see if there were any relationships between the most important stats, rebounds with the winrate of a team. As seen in Figure 6, we plotted, from highest win rate on the left to lowest win rate on the right, with their average amount of rebounds per game. We can see that there is a slight linear relation as the fitted line decreases in the amount of rebounds as the win rate decreases. This aligns well with our results in Feature Importance as we can conclusively say that rebounds is the most significant in-game stat in the NBA.

Figure 7.



Lastly, when determining whether personal fouls were the least significant factor in predicting the outcome of the game, we predicted that there would be little to no linearity. As seen in Figure 7, we can see that our prediction was true as the linear fit is almost constant. This also aligns well with our results in Feature Importance as we can conclusively say that personal fouls is the least significant in-game stat in the NBA.

5. Conclusion

From the beginning, our goal was to determine which in-game stat in the NBA contributed most to winning the game. Our hope was to find some focus point of the game for teams to increase their win rate. In our first step, we were able to determine that all the variables were statistically significant from performing T-tests. Moving into training and testing our various models, we observed that our models trained with the variables were able to yield high accuracy scores of around 77%. This allowed us to check which were the most important variables in making the predictions, which we found to be rebounds, and the least significant to be personal fouls. Finally, we checked whether those variables had a relationship with a teams' win rate and found that rebounds, the most important feature across all the prediction models, has a linear relationship with the win rate. Thus, we are able to conclude that rebounds are the most important statistic in winning a game in the NBA.

6. Limitations

Problems

At the start, we were unsure of which variables to include so we included all of them. This meant that we had points, field goals made, free throws made, and assists in our data. As we started training models, we found that the accuracy scores were very high, 98%-100%. It was then that we realized that these variables were extremely biased as they were the actual factor that determined who won or lost. As a result, we had to remove them in the Data Pre-processing step.

More Time

After further analyzing all the data in the original game.csv provided by Kaggle, we found that the games dated back to 1982. This means that there are teams that are not in the league anymore that are still included in our data, or the data is strictly outdated. If we had more time, we would have liked to perform the same tests on more select years to see if there were any changes in stat importance over the years.

Retrospect

This project ended up being more difficult and time consuming than expected. In retrospect, we would have definitely started much sooner.