Predicting Individual NBA Home Team Game Outcomes Using Team Statistics

Travis Knoche: tk6vw
Rehan Merchant: rm2bt
Drake Wagner: dbw2tn

Abstract

The goal of our project was to create an algorithm that can accurately predict the outcome of an NBA game based solely on statistics from each team (besides home points vs away points). Future game predictions are important to teams wanting to understand which statistics are the most important in predicting wins in the league, as well as aiding sports analysts, sports bettors and oddsmakers before events take place. Before beginning our analysis, we aimed for an algorithm that would be able to correctly predict the winning team 65% of the time. We used Apache Spark as our data processing engine for the project, while we wrote our code in Python. The data is a Kaggle dataset logging NBA game statistics for every game between the 2004 and 2020 seasons, consisting of many different player, game, and team statistics. We uploaded this data into Spark, and created logistic regression, random forest, and linear support vector machine models. We performed different pre-processing methods on the data, and trained the models on 70% of the data, while reserving the remaining 30% for validation. We found that the logistic regression model was the most accurate in predicting home team wins with an accuracy of 0.8717. (Note: Before the presentation, we made an error in the variables we kept in the model, and initially included away team points as well as home team points, so each model had near perfect accuracy. After presenting, we re-ran the models and have adjusted the results here accordingly). We conclude that we are able to moderately accurately predict the outcome of an NBA game if given the major statistics aside from away team points.

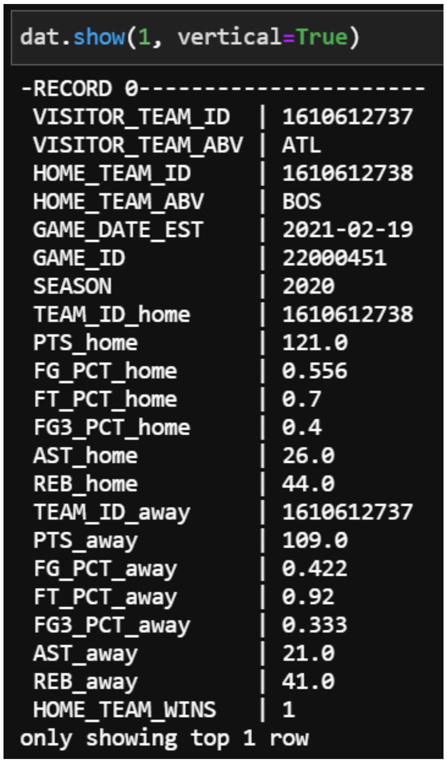
Introduction

Out of all the most commonly played and marketed sports, basketball is the highest scoring. As a versatile team-based competition, a large number of statistics go into a basketball team's success. Within the National Basketball Association, the ability to recruit valuable players is one of the largest factors in running a successful franchise and winning championships. Within this recruiting process, not every player has the same skillset. Some athletes are more defense-oriented, while others are better at offense. Some prioritize three-point shooting ability, while others shoot more two-pointers. With every player having a unique skill set, it could be difficult to decide who to recruit or draft. Additionally, while individual players contribute to a team's performance, the end result usually comes down to how the team as a whole does. The objective of this project is to illuminate which statistics are most valuable in predicting whether a team wins or loses a basketball game.

Data and Methods

The particular dataset used in this project was chosen because of the large amount of different statistics as well as the size of the data. Seasons between 2004 and 2020 were included, giving us a good amount of data to train and test our model with.

We began by looking through the datasets and deciding on which columns would be useful in our project. Five different csv files were included, labeled as Games, Games Details, Players, Ranking, and Teams. Because we focus our project on entire team win percentages, we quickly realized that we did not need any columns from the players file. Ranking mostly held columns related to each team's current ranking and winning percentage throughout the league, by day. We did not include any of these since we don't focus on change over time. The Games Details and Teams csv files ended up being the most useful, since they included varying statistics that were likely to correlate with whether a team wins or loses, such as rebounds, assists, blocks, shot percentage, and number of points. It also included basic data on whether the team wins or loses the game, which teams were playing, and whether it was a home game or away game.

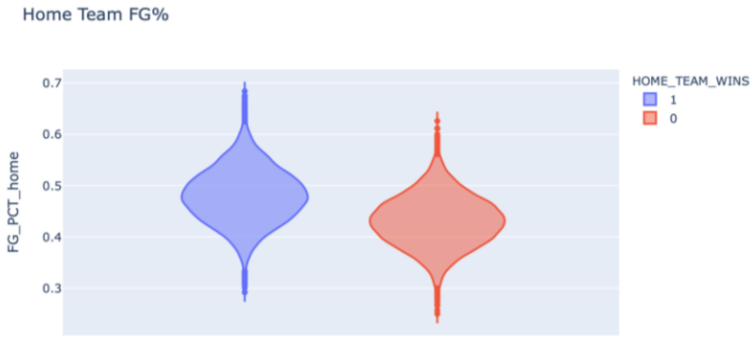


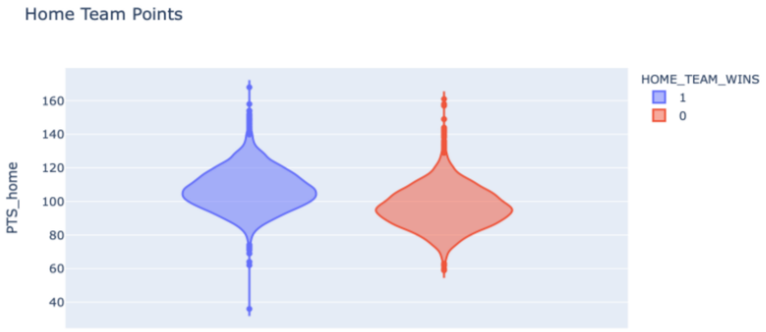
```
dat.show(1, vertical=True)
```

```
-RECORD 0----------------------
 VISITOR_TEAM_ID   | 1610612737
 VISITOR_TEAM_ABV  | ATL
 HOME_TEAM_ID      | 1610612738
 HOME_TEAM_ABV     | BOS
 GAME_DATE_EST     | 2021-02-19
 GAME_ID           | 22000451
 SEASON            | 2020
 TEAM_ID_home      | 1610612738
 PTS_home          | 121.0
 FG_PCT_home       | 0.556
 FT_PCT_home       | 0.7
 FG3_PCT_home      | 0.4
 AST_home          | 26.0
 REB_home          | 44.0
 TEAM_ID_away      | 1610612737
 PTS_away          | 109.0
 FG_PCT_away       | 0.422
 FT_PCT_away       | 0.92
 FG3_PCT_away      | 0.333
 AST_away          | 21.0
 REB_away          | 41.0
 HOME_TEAM_WINS    | 1
only showing top 1 row
```

A python file was created specifically for combining the data that we needed. In that file, we read in the csv files, merged Game Details and Teams by their shared column TEAM_ID, reworded the team names and abbreviations to make it easier to understand, and finally dropped the unneeded columns. This newly revised dataset was the one we used for our project. To the right is an image of the columns included in this merged dataset. Keep in mind that several more of these columns were dropped during our data selection, including the duplicates in team id.

Exploratory data analysis showed several correlations in the data before we ran any tests. Many of these results were self explanatory and were what we were expecting, such as a positive correlation between assists and points. Interestingly, field goal percentage had the highest correlation with predicting if the home team wins, even more so than number of points, which was the second highest correlation. The graphs of these results are shown below.

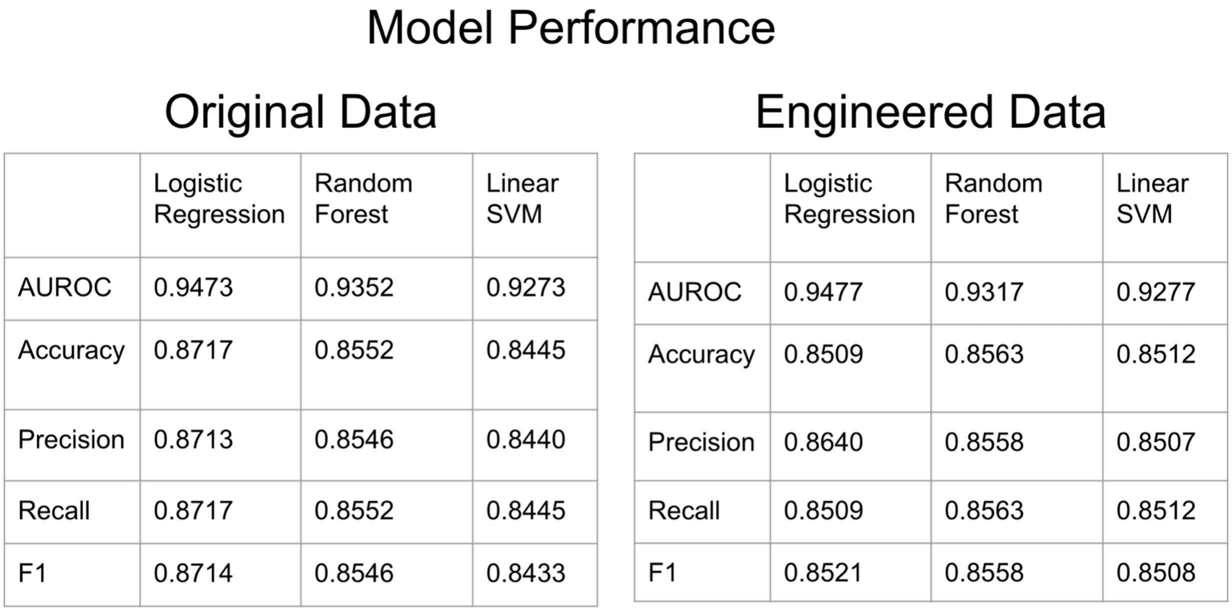




All of our code was built and run through PySpark, except for several small instances of exploratory data analysis, variable importance extraction, recall-threshold plotting, and the file conversion code which were done with base Python or Pandas. Within the Spark framework, many different model types and feature variables were created in order to determine which combination would work best in predicting the outcome of a certain game. Correlations in the data in relation to whether the game was won or not were looked at and used in creating feature variables. For example, field goal percentage turned out to be the highest correlated statistic with whether the game was won or not, so we began by transforming this data to see if it could more accurately predict the outcome. We created two variables, FG(field goal)%^2 and PTS_home^2, that we thought might have a quadratic effect, but the best performing model was the logistic regression with no engineered features.

Next, a 70-30 training-test data split was applied to the original data as well as the engineered data. Vector assembly and standard scaling were run through spark to scale the unit variance using the column summary statistics. Data was then pipelined into varying model types, including linear regression and random forest analysis. We conducted cross validation of hyperparameters with k=5 folds, to select the final parameters for each model. After these steps, we were able to look at how our models performed, and begin to select our most appropriate model.

Results



Model Performance

Original Data

	Logistic Regression	Random Forest	Linear SVM
AUROC	0.9473	0.9352	0.9273
Accuracy	0.8717	0.8552	0.8445
Precision	0.8713	0.8546	0.8440
Recall	0.8717	0.8552	0.8445
F1	0.8714	0.8546	0.8433

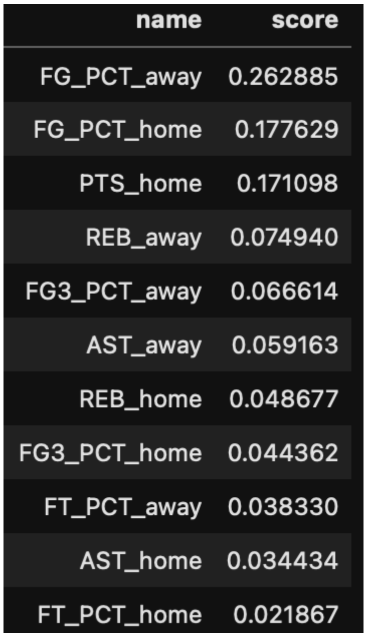
Engineered Data

	Logistic Regression	Random Forest	Linear SVM
AUROC	0.9477	0.9317	0.9277
Accuracy	0.8509	0.8563	0.8512
Precision	0.8640	0.8558	0.8507
Recall	0.8509	0.8563	0.8512
F1	0.8521	0.8558	0.8508

Our results show that it is entirely possible to create an accurate prediction model of games using season and player statistics from past seasons. Our best models used the standard statistics included in the original files, bypassing the need for feature variables.

Of our three listed models above, Random Forest and Logistic Regression with the original data were the all-around best performing models. Each model was run through cross-validation in order to fine tune parameters.

Random Forest and Logistic Regression are both attributed with very high accuracy and AUROC. As for precision, recall, and F1, Logistic Regression with the original data has the upper edge over Random Forest. We believe that Logistic Regression gives us the best chance of giving an accurate and precise prediction to the outcome of each game, although Random Forest would also work very well if needed. The Random Forest model is also informative, with its extracted variable importance scores shown to the right.



name	score
FG_PCT_away	0.262885
FG_PCT_home	0.177629
PTS_home	0.171098
REB_away	0.074940
FG3_PCT_away	0.066614
AST_away	0.059163
REB_home	0.048677
FG3_PCT_home	0.044362
FT_PCT_away	0.038330
AST_home	0.034434
FT_PCT_home	0.021867

Conclusions

The results from our analysis show that the Logistic Regression model with the original data set predicted wins the best. The accuracy of our model on test data was 0.8717. We can confidently predict if the home team will win or not based on the statistics the team averages over the season. This will help team executives have an idea on how the team is performing at home and how they can expect to perform over the course of the season at home. This model also gives insight into how management can improve the team This shows that given certain inputs we can accurately predict outcomes when a home team plays. This model will be really useful in evaluating rosters of NBA teams.

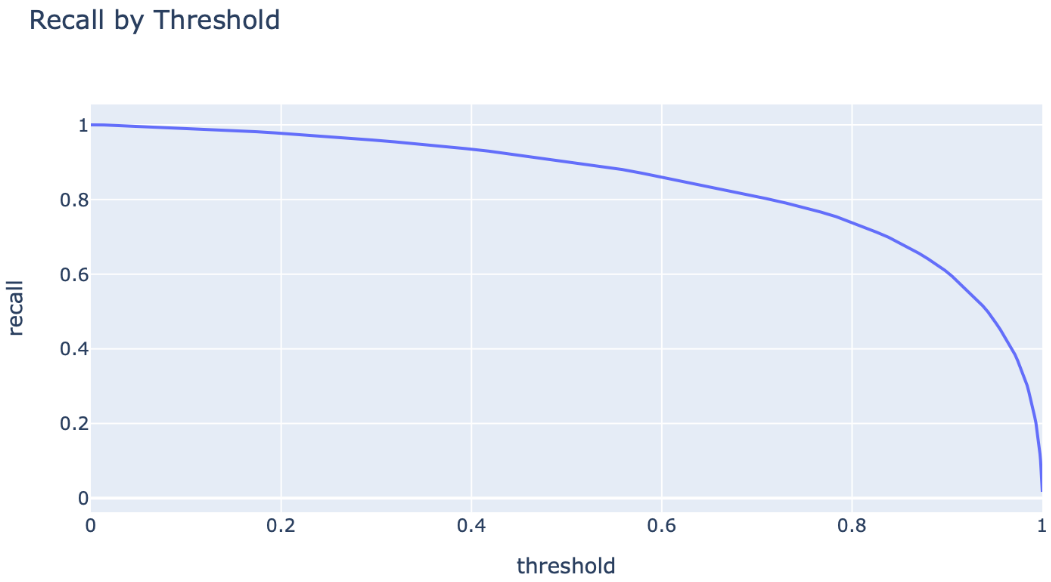
Our analysis can be used by NBA front office executives as a tool to help evaluate the rosters of their team. Since this tool shows the likelihood of the home team winning, it will give executives an insight into how their team, currently constructed with the statistics as of last season, will do. This allows the executives to narrow down key areas in order to improve the team. Executives will have a head start in creating draft boards based on needs and targeting free agents that may help bring certain averages up, helping improve the team's chances of winning more games at home, ultimately leading to a championship.

Team executives will want to use this model to help assemble the team. Based on the model, teams will want to target players with a high field goal percentage who also put up a high amount of points. Although this is a very common concept in basketball, our model will be able to help teams see what benchmark they need to reach in order to get that home win. In turn, teams will have an idea on what kind of players they need to target in order to help them reach that benchmark.

To further improve our project, our team would like to match up certain teams and use the model to predict which team would win in a head to head matchup. In order to do this, we would have to include player statistics since teams change season to season. This would be very manageable since individual statistics are included within the dataset. After creating a new model including player statistics, we would aggregate the statistics based on the team they are on as of the start of the new season. The biggest challenge in this would be to account for rookies since they do not have historical data in the NBA.

This new model would be used to help teams fill out rosters by identifying weak spots and which areas the team would need to improve in order to get more wins, increase TV viewership/broadcasting contracts, and ultimately make more money. This would help narrow down the list of players that would fit the role the team is looking for within free agency, the draft and at the trade deadline. Being able to target players early on will give the team an advantage in getting the player at the right price or at the right compensation without giving up too many assets in a trade. Another interesting use-case is that of sports bettors and oddsmakers in the gambling world. This information could lead to more informed betting as well as profit maximization by the oddsmakers.

Also, in the future, we would like to conduct thorough threshold analysis, as supported by the chart below. This could aid in further model tweaks, and possible features to add, such as certain teams' histories against other teams at different points in the season. We could potentially create a more dynamic, team-specific model. We also could create a model for each team, for each year. This could further be analyzed in conjunction with each team's roster, salary cap utilization, and other considerations.


Recall by Threshold
recall
threshold